



PROYECTO DE SISTEMAS INFORMÁTICOS
Curso 2009/2010

Juegos educativos en dispositivos móviles

Autores:

Álvaro Villoria Sáez,
Guillermo Martín Pérez,
Juan Manuel de las Cuevas Caminero

Directores:

Baltasar Fernández Manjón,
Iván Martínez-Ortiz

Se autoriza a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

Juan Manuel Cuevas Caminero

Guillermo Martín Pérez

Álvaro Villoria Sáez

Agradecimientos

Deseamos expresar nuestro más sincero agradecimiento por el apoyo de todas las personas que nos han ayudado durante la realización del proyecto.

Agradecemos la ayuda recibida por los integrantes del departamento de investigación e-UCM que nos han guiado con su experiencia, muy especialmente a Javier Torrente y Eugenio Marchiori por soportarnos cada semana.

Debemos también agradecer a Navarro Romero y Héctor Montoya por la ayuda recibida en la creación de los recursos artísticos de la aplicación desarrollada.

Por último agradecer el apoyo de nuestros compañeros y amigos por haber estado a nuestro lado durante la realización del proyecto.

Juegos educativos en dispositivos móviles

Autores:

Álvaro Villoria Sáez,

Guillermo Martín Pérez,

Juan Manuel de las Cuevas Caminero

Directores:

Baltasar Fernández Manjón,

Iván Martínez-Ortiz

Índice

| | |
|--|-----|
| Resumen..... | 11 |
| Abstract..... | 12 |
| Acerca de este documento | 13 |
| | |
| 1. Introducción | 15 |
| 2. Planteamiento del proyecto..... | 17 |
| 2.1. Motivación del proyecto | 17 |
| 2.2. Estudio de plataformas..... | 19 |
| 2.3. Objetivos y desafíos..... | 28 |
| 2.4. Metodología y planificación | 32 |
| 3. Contexto y trabajo relacionado | 37 |
| 3.1. M-Learning | 37 |
| 3.2. Juegos educativos | 38 |
| 3.3. Juegos en dispositivos móviles | 40 |
| 4. Adaptación de juegos educativos a dispositivos móviles | 43 |
| 4.1. Interacción de usuario en dispositivos móviles | 44 |
| 4.2. Distribución de juegos educativos en entornos m-Learning..... | 71 |
| 4.3. Nuevas tecnologías en dispositivos móviles, GPS y cámara de fotos | 74 |
| 5. Análisis de las plataformas escogidas..... | 77 |
| 5.1. Análisis de eAdventure | 77 |
| 5.2. Análisis de Android | 83 |
| 6. Análisis del sistema | 97 |
| 7. Diseño e Implementación | 119 |
| 7.1. Motor de juegos para Android | 119 |
| 7.2. Adaptación del editor de juegos de eAdventure | 126 |
| 7.3. Repositorio de juegos eAdventure | 128 |
| 8. Optimizaciones del sistema | 129 |
| 9. Caso de uso | 135 |
| 9.1. Descripción educativa..... | 136 |

| | | |
|-------|--|-----|
| 9.2. | Guión del juego | 137 |
| 9.3. | Planificación y desarrollo del juego | 138 |
| 10. | Conclusiones | 143 |
| 10.1. | Contribuciones | 144 |
| 10.2. | Trabajo futuro..... | 147 |
| 11. | Referencias | 151 |
| 12. | Bibliografía..... | 153 |

Resumen

Las herramientas de autoría de juegos educativos permiten la creación de juegos educativos para ordenadores y navegadores Web reduciendo el coste y tiempo de desarrollo. Sin embargo, el desarrollo de juegos ha cambiado en los últimos años, con un aumento significativo de los juegos creados para dispositivos móviles. Se ha contribuido a una herramienta de autoría de juegos educativos que permite la adaptación de los juegos actuales a las plataformas móviles y la creación de nuevos tipos de juegos educativos que aprovechan las nuevas tecnologías que proporcionan los dispositivos móviles (por ejemplo, GPS, cámara de fotos). Con la inclusión de estas nuevas tecnologías los juegos educativos en dispositivos móviles permiten el aprendizaje ubicuo, proporcionando alto contenido interactivo, en diferentes situaciones. Se ha desarrollado una ampliación de la plataforma de juegos educativos eAdventure haciendo uso de la plataforma móvil Android TM.

Palabras clave: **Educación, M-Learning, Android, Videojuegos, Evaluación, Adaptación, Editor, Motor, Móvil**

Abstract

Educational game authoring tools allows the creation of educational games for personal computers and web-browsers reducing costs and development cycles. However, gaming has changed in the last few years, with a significant increase of games developed for mobile devices. We have developed an extension of an existing educational game authoring tool that allows the adaptation of current games to mobile platforms and the creation of new types of educational games taking advantage of new features provided by mobile devices (e.g. GPS, camera). With the inclusion of these new features, mobile educational games allow for ubiquitous learning by providing highly interactive and engaging contents with educational value in different situations. This development has been done extending the eAdventure educational game platform and using Android™ as the mobile platform.

Keywords: Education, M-Learning, Android, Videogames, Assessment, Adaptation, Editor, Engine, Mobile

Acerca de este documento

El presente documento describe el trabajo realizado durante el proyecto de Sistemas Informáticos por Álvaro Villoria Sáez, Guillermo Martín Pérez y Juan Manuel de las Cuevas Caminero, bajo la dirección de Baltasar Fernández Manjón e Iván Martínez-Ortiz, director y miembro respectivamente del grupo <e-UCM> de investigación en el campo de e-Learning de la Universidad Complutense de Madrid.

El documento sigue una estructura sencilla basada en el proceso seguido a lo largo del proyecto. En primer lugar se describe el proyecto desde un punto de vista global y se introducen conceptos básicos de la temática de éste para continuar entrando en detalle según avanza el documento.

Posteriormente se analiza de manera breve el estado del campo de los videojuegos educativos y el aprendizaje mediante dispositivos móviles, con el fin de contextualizar mejor los problemas en los que se centra el proyecto y situarnos en un punto a partir del cual poder aportar soluciones interesantes. Más adelante se entra en detalle sobre la aportación realizada, soluciones teóricas e ideas interesantes en la adaptación de videojuegos educativos a dispositivos móviles. A continuación se aborda el grueso de la parte técnica donde se describe el proceso de análisis, diseño e implementación del sistema que satisface los objetivos establecidos.

Por último se proporciona una sección de conclusiones, discusión de las principales aportaciones del proyecto así como reflexión del trabajo futuro, referencias y bibliografía.

1. Introducción

El proyecto surge como respuesta al creciente interés en los últimos años que existe en la aplicación de videojuegos y dispositivos móviles en contextos educativos. No obstante, existen multitud de barreras que obstaculizan la generalización de esta nueva forma de aprendizaje, tales como el elevado coste de desarrollo de los videojuegos, o la complejidad que conlleva su introducción en el flujo educativo. Asimismo, debido a lo reciente de este interés, no existen todavía fórmulas pedagógicas ampliamente aceptadas en la comunidad científica que permitan aprovechar eficazmente el potencial de estas tecnologías.

Del análisis anterior surge la principal motivación del proyecto: proporcionar una herramienta de autoría que facilite la creación de videojuegos educativos para distintos entornos y plataformas, con especial interés en la compatibilidad con dispositivos móviles. De esta manera el proyecto consta de las siguientes líneas de actuación:

- Proponer estrategias que permitan el despliegue y ejecución de videojuegos educativos en plataformas móviles.
- Analizar las nuevas características hardware que ofrecen los dispositivos móviles y proponer un modelo de uso de los mismos para videojuegos educativos.
- Crear o adaptar una herramienta de edición de videojuegos educativos que minimice tanto en tiempo como en coste, el desarrollo y despliegue de éstos para dispositivos móviles.

2. Planteamiento del proyecto

El siguiente apartado introduce brevemente el motivo por el que resulta interesante realizar un proyecto de estas características. Se aborda la motivación del proyecto así como los objetivos principales que se fijaron inicialmente y los desafíos que conlleva su realización.

2.1. Motivación del proyecto

Durante las últimas décadas está aumentando el interés en la idea de cómo los ordenadores y los videojuegos pueden aportar experiencias útiles en el aprendizaje (Van Eck 2006). Que los usuarios puedan libremente explorar el contenido del juego, razonar y sacar conclusiones propias hace que los videojuegos puedan ayudar a la adquisición de conocimiento profundo del dominio de estudio (Aldrich, 2005).

Por otro lado también ha surgido un amplio interés en la aplicación de dispositivos móviles en contextos educativos dado su elevado potencial, tal y como podemos ver en la literatura reciente del campo (Squire, 2004). Este interés se remonta prácticamente a los orígenes de la tecnología móvil, dando lugar a una corriente conocida como *Just-In-Time Learning*, es decir, aprender en el sitio y el momento requerido (Gee, 2003). La idea es que el alumno pueda acceder al contenido educativo justo en el momento y lugar que sea necesario independientemente del dispositivo que tenga disponible. No obstante, mientras que esto es factible para ciertos contenidos educativos que pueden desplegarse en distintas plataformas (información HTML, documentos, material multimedia, etc.) otros contenidos más sofisticados como son los videojuegos requieren una adaptación específica al dispositivo concreto, incrementando en buena proporción el coste de desarrollo y mantenimiento.

Es por ello que surge la necesidad de proporcionar soluciones para la creación de videojuegos educativos que permitan su despliegue tanto en entornos de escritorio como

en dispositivos móviles sin poner en riesgo el coste de desarrollo ni la mantenibilidad de los mismos.

Juegos educativos en dispositivos móviles

Tanto los juegos educativos como el aprendizaje a través de dispositivos móviles proporcionan ventajas dentro del ámbito del aprendizaje electrónico.

En este caso el proyecto se centra en combinar ambas ramas aportando valor a los juegos educativos sacando el máximo partido de la tecnología móvil. La combinación de ellas hace posible nuevas vías de aprendizaje y nuevos paradigmas en el diseño y creación de videojuegos.

Un ejemplo donde los juegos educativos en dispositivos móviles pueden mejorar la experiencia de aprendizaje se da en escenarios de entrenamiento. En estos casos los alumnos pueden explorar el problema dentro de un ambiente seguro, el del juego educativo, estando situados en el entorno real beneficiándose al mismo tiempo de ambos “mundos”.

Como caso real imaginemos el escenario de un trabajador sin experiencia que se incorpora a la cadena de montaje de una fábrica de automóviles. El proceso está totalmente definido y se debe seguir una serie de pasos para completar la construcción de los vehículos satisfactoriamente y de manera segura. Con un juego educativo desplegado en un dispositivo móvil el usuario puede desplazarse por una cadena de montaje interactuando en un entorno seguro, el del mundo virtual, mientras observa como es el funcionamiento real del proceso.

Por otro lado estimular la interactividad de los usuarios en el juego puede ayudar a mejorar la actividad educativa influyendo en el aprendizaje a largo plazo. Las nuevas tecnologías incorporadas en los terminales móviles actuales amplían las posibilidades de interacción entre dispositivos y usuarios. En el caso del geoposicionamiento global (GPS) y la comunicación de campo cercano (Bluetooth o WiFi) contribuyen al aumento de la inmersión e integración de los jugadores en el juego.

2.2. Estudio de plataformas

En el siguiente apartado se realiza un estudio de las plataformas que pueden aportar una base sólida para el desarrollo del proyecto. En primer lugar se analizan algunas de las herramientas de creación de videojuegos más relevantes y posteriormente se estudian distintas plataformas móviles.

Plataforma creación y edición de videojuegos

Dentro de las principales líneas de actuación del proyecto, se definió proporcionar una herramienta de autoría que facilitase la creación de videojuegos educativos para distintos entornos y plataformas, incluyendo la posibilidad de desarrollar para una plataforma móvil.

Debido a la magnitud del proyecto se consideró innecesario desarrollar la herramienta por completo, puesto que existen cantidad de ellas que pueden ser mejoradas para dar soporte a móviles.

Para ello se estudiaron distintas alternativas existentes en el mercado teniendo en cuenta la viabilidad e interés de realizar una ampliación en alguna de ellas. El criterio de elección de la plataforma se basa en los siguientes puntos clave:

- Se valoran las plataformas de código libre que permitan la distribución gratuita, tanto de la plataforma, como del contenido creado a partir de ella.
- La creación de videojuegos debe ser sencilla e intuitiva y no debe requerir conocimiento técnico avanzado en programación ya que el proyecto está dirigido a usuarios inexpertos como tutores y alumnos.
- Son valoradas aquellas plataformas que disponen de herramientas orientadas al campo de la educación que faciliten la creación de juegos educativos a los tutores.

Blender

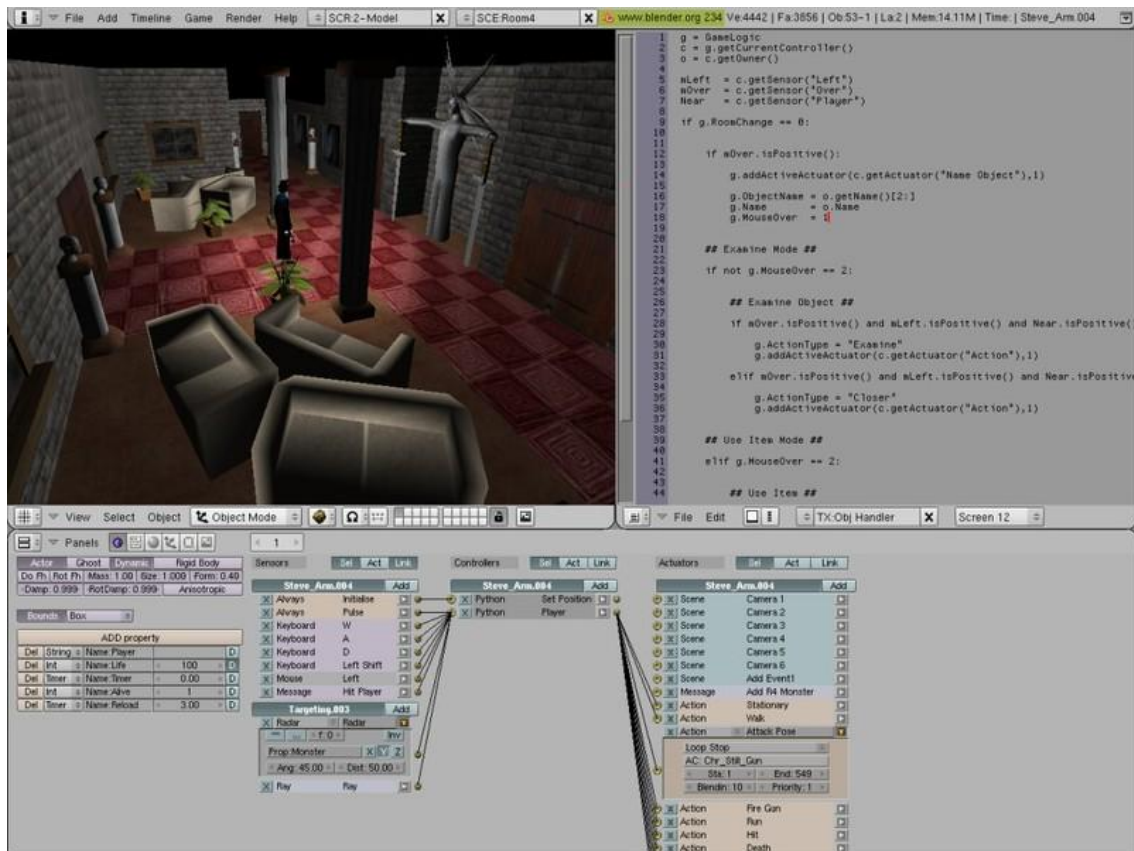


Ilustración 1 - Entorno de edición de juegos de la plataforma GameBlender

Se trata de una plataforma de creación de videojuegos especializada en juegos de tres dimensiones. Consta del editor *GameBlender* y de un motor que permite la ejecución de los juegos en ordenadores de escritorio. Es de código libre y cuenta con documentación abundante disponible a través de tutoriales *on-line*.

A pesar de todo, la curva de aprendizaje de la herramienta puede ser un inconveniente de cara al propósito del proyecto ya que el desarrollo de los videojuegos requiere de un conocimiento y esfuerzo elevado debido a la complejidad de la creación de los recursos en tres dimensiones.

Adventure Maker

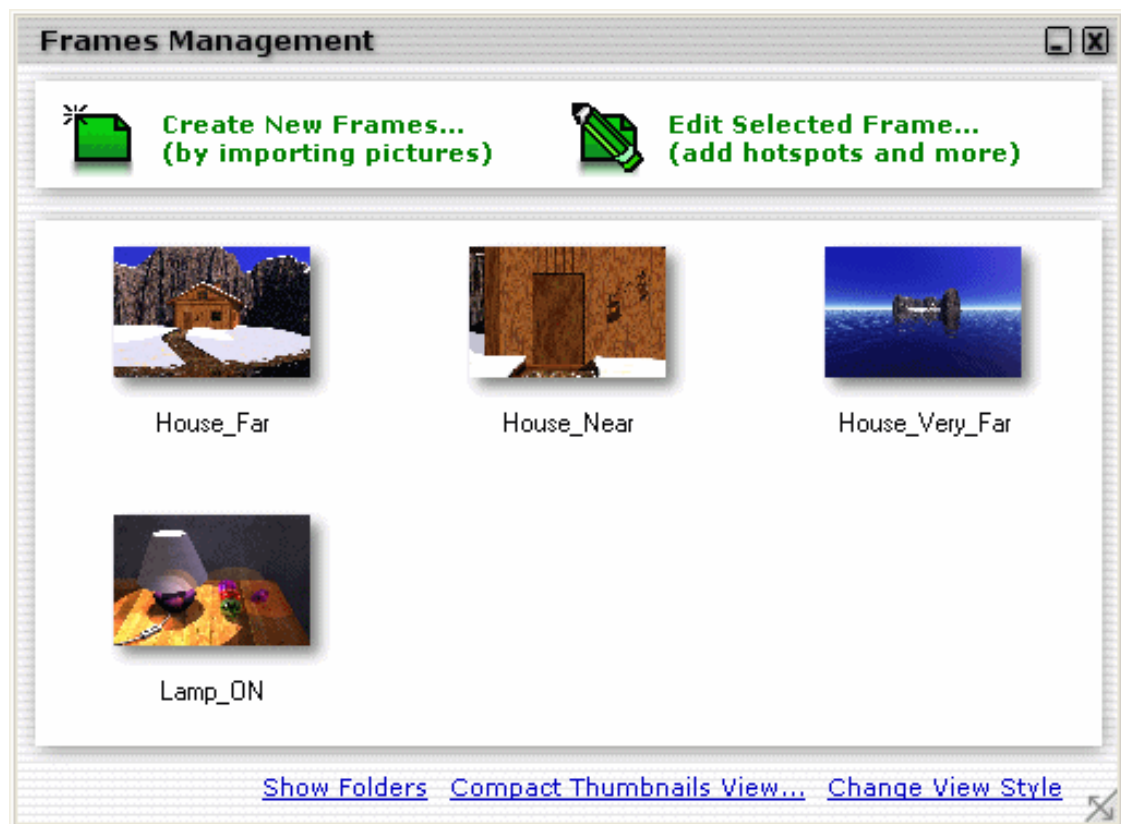


Ilustración 2 - Herramienta de creación de juegos de Adventure Maker

Adventure Maker permite crear videojuegos de tipo *point-and-click* y desplegarlos en distintas plataformas. Incluye la posibilidad de exportarlos para ordenadores personales además de dispositivos móviles y consolas portátiles como *iPhone* y *PlayStation Portable*. Otra característica interesante de la plataforma es la capacidad de incorporar extensiones que amplíen la funcionalidad de los juegos creados a partir de ella. Sin embargo se requieren conocimientos de un lenguaje propio de *scripting* que dificulta la elaboración de los juegos de cara a un entorno educativo. Además el acceso a todas las funcionalidades del sistema está limitado a su versión de pago.

Game Salad



Ilustración 3 - Herramienta de edición de juegos de la plataforma GameSalad

Game Salad permite el diseño de videojuegos mediante un sistema gráfico basado en *drag and drop*, siendo éste un sistema intuitivo para la creación de videojuegos. El despliegue de juegos está delimitados a la Web (tecnología Flash) y a sistemas Apple, tanto de escritorio como móviles en iPhone, iPod Touch y iPad. La plataforma no está orientada a ser usada en un entorno educativo y es una plataforma propietaria y de pago.

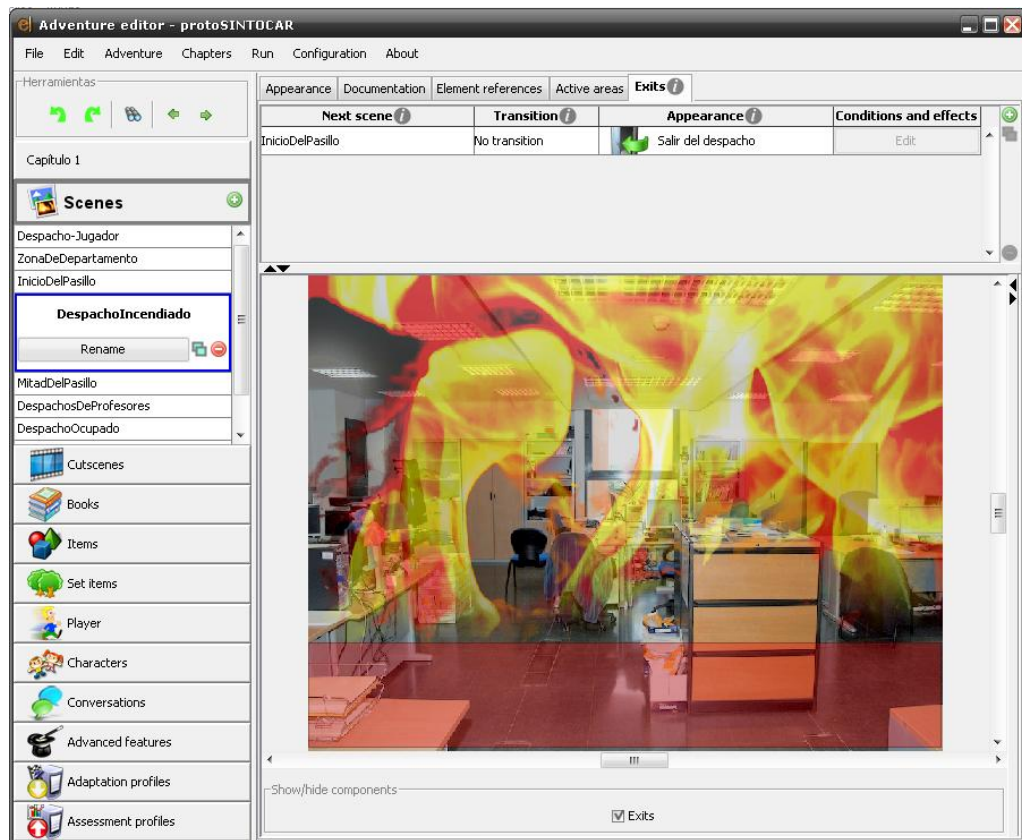


Ilustración 4 - Herramienta de edición de juegos de la plataforma eAdventure

eAdventure es una plataforma de código libre que permite crear y distribuir juegos educativos en ordenadores personales y en la Web. Consta de un editor y motor de aventuras gráficas. La creación de los videojuegos es intuitiva y sigue la filosofía de *point-and-click*, de manera que el usuario no necesita tener conocimientos de programación. La plataforma está orientada a su uso en un ámbito educativo con herramientas como la evaluación y adaptación del contenido a los usuarios. Por otro lado la plataforma cuenta con escasa documentación.

Elección de la plataforma de autoría de juegos

eAdventure es la plataforma escogida ya que se ajusta mejor al criterio de elección establecido. En primer lugar eAdventure es una plataforma libre que puede ser adaptada a las necesidades del proyecto. Además es la única plataforma orientada a ser usada en un entorno educativo. Por otro lado, eAdventure pertenece al grupo de investigación eUCM de la Universidad Complutense de Madrid, siendo posible la comunicación directa con los desarrolladores de la plataforma supliendo de alguna manera la carencia de documentación existente.

Plataforma móvil

El mercado de los dispositivos móviles está en pleno proceso de evolución. Existe gran variedad de configuraciones hardware y de sistemas operativos disponibles. Por ello antes de optar por una plataforma para el desarrollo del proyecto se deben tener en cuenta los siguientes factores:

- Elegir una plataforma con cierta relevancia y proyección de futuro de cara a mantener el proyecto estable a largo plazo.
- Asegurar la compatibilidad de la plataforma con el mayor número posible de dispositivos móviles de última generación del mercado.
- Disponer de acceso libre a la documentación y las herramientas de desarrollo de la plataforma.

En primer lugar se estudia el estado del mercado de los dispositivos móviles con el fin de determinar las plataformas más relevantes. Para ello a continuación se muestran dos gráficas basadas en el tráfico Web de cada sistema operativo móvil a nivel mundial, recolectado por la empresa AdMob¹ durante el mes de Febrero de 2010.

¹ <http://metrics.admob.com/2010/03/february-2010-mobile-metrics-report/> (02/Abril/2010)

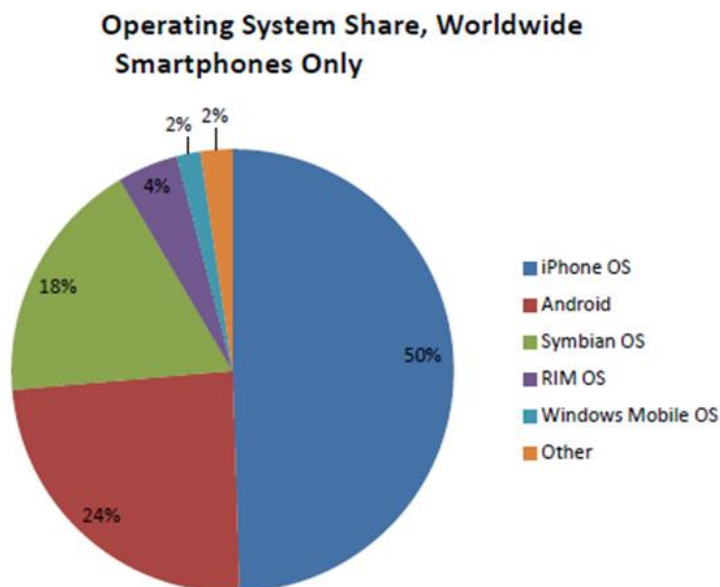


Ilustración 5 - Uso de sistemas operativos a nivel mundial. Febrero 2010 (AdMob)

Como se puede apreciar en la Ilustración 5 existen tres plataformas predominantes en el mercado de los móviles de última generación. En primer lugar se sitúa iPhone como la plataforma más extendida abarcando la mitad de todos los dispositivos de la muestra. El segundo lugar lo ocupa el sistema operativo Android que está presente prácticamente en uno de cada cuatro dispositivos analizados. Por último Symbian es el sistema usado por el 18% de los dispositivos.

A pesar de las grandes diferencias entre los porcentajes de uso, es importante evaluar otros factores más allá de la popularidad en un instante de tiempo determinado. Por ello se muestra un gráfico (Ilustración 6) que describe la evolución del tráfico Web proveniente de los dispositivos de última generación a nivel mundial. Se puede apreciar que iPhone es una vez más la plataforma más popular tras un crecimiento que contrasta con la pérdida de mercado de los móviles Symbian. Asimismo Android se sitúa en el segundo lugar experimentando un crecimiento gradual durante los últimos 16 meses y superando la cuota de Symbian.

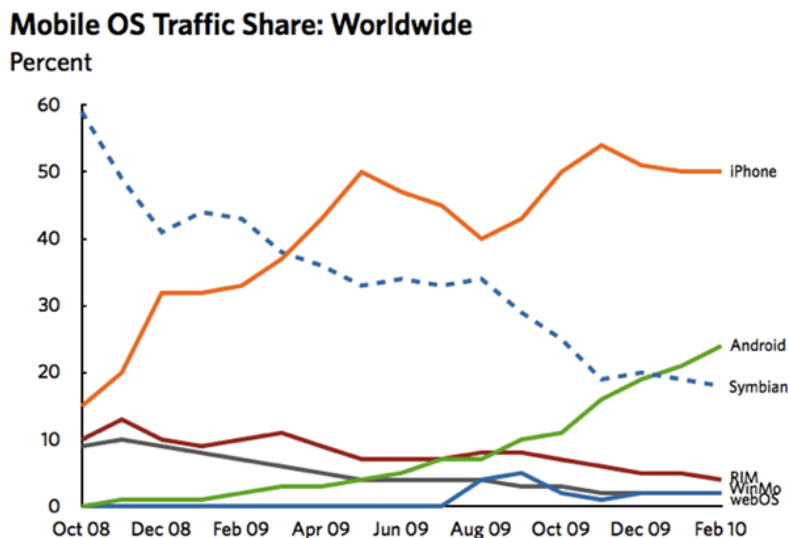


Ilustración 6 - Tráfico Web en móviles de última generación a nivel mundial. Febrero 2010

Teniendo en cuenta los datos analizados, se descartan las plataformas con menor proyección de futuro, para así centrar el análisis sobre las plataformas de iPhone y Android.

iPhone OS

iPhone es la plataforma más extendida en el mercado de los dispositivos móviles de última generación y una de las plataformas con mayor proyección de futuro. Dispone de documentación técnica abundante y actualizada. La plataforma proporciona herramientas para la creación de aplicaciones para distintos dispositivos móviles *iPhone*, *iPad* y *iPod Touch*. Sin embargo, iPhone OS se trata de una plataforma cerrada. Apple ofrece un conjunto de herramientas para el desarrollo de aplicaciones que utiliza el lenguaje Objective-C. El conjunto de desarrollo sólo está disponible para ordenadores fabricados por Apple. Además el único medio de distribución de aplicaciones es a través de su tienda de aplicaciones (iTunes App Store).

Android

Sistema operativo de código libre basado en el núcleo de Linux. La plataforma Android ofrece un conjunto de herramientas de desarrollo disponible para ordenadores con sistema operativo Windows Linux y Mac. Al igual que iPhone OS, dispone de documentación abundante y actualizada. Las aplicaciones para Android pueden escribirse tanto en lenguaje Java como en C/C++. El sistema operativo es multitarea y la distribución de las aplicaciones se puede realizar a través de la tienda de aplicaciones *Android Market* y directamente a través de los ficheros de aplicaciones Android. Más allá de sus características técnicas, Android es utilizado como sistema operativo en una amplia gama de dispositivos pertenecientes a distintos fabricantes.

Elección de la plataforma móvil.

Tras haber analizado las dos plataformas más relevantes del mercado de los dispositivos móviles se decide basar el proyecto en la plataforma móvil Android. iPhone es la plataforma más popular en la actualidad, sin embargo Android tiene cada vez más presencia en el mercado de los dispositivos móviles. Además es una plataforma libre, de manera que se adecua más a la filosofía del proyecto y da soporte a las nuevas tecnologías presentes en los móviles de última generación, siendo interesante de cara al desarrollo del proyecto.

2.3. Objetivos y desafíos

A continuación se detallan los principales objetivos y desafíos del proyecto teniendo en cuenta la decisión de contribuir a la plataforma eAdventure.

Objetivos del proyecto

El principal objetivo del proyecto se centra en proporcionar soporte a la creación de juegos eAdventure para Android.

eAdventure cuenta con un proceso de creación y distribución de juegos educativos de tipo aventura gráfica, sencillo y multiplataforma.

Como trabajo del proyecto se plantea revisar y ampliar el proceso en sus distintas fases.

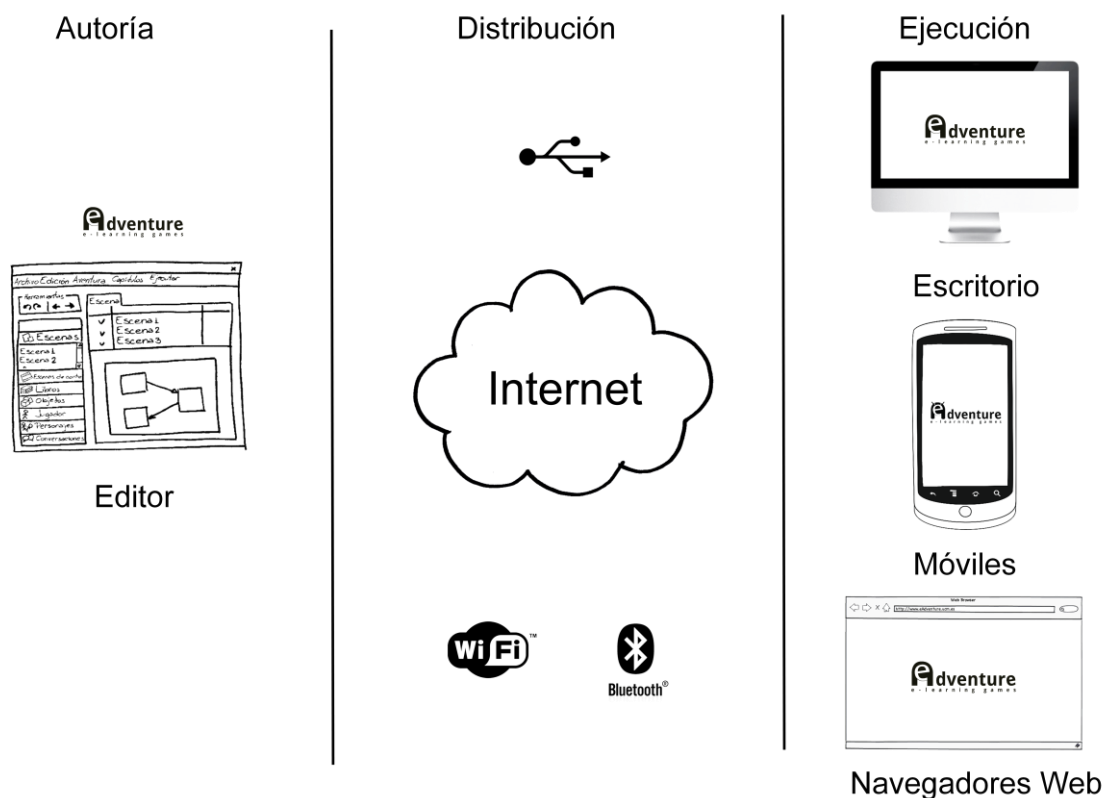


Ilustración 7 - Proceso de creación, distribución y ejecución de juegos eAdventure

Autoria – Editor de eAdventure

El editor de eAdventure debe ser revisado y ampliado para incluir el soporte de juegos para Android. Los objetivos planteados inicialmente en la fase de edición son los siguientes:

- La creación de juegos debe permitir su despliegue en las plataformas de escritorio, Web y móviles Android.
- Los juegos anteriormente creados con la plataforma deben ser compatibles con dispositivos móviles Android (Compatibilidad hacia atrás).
- El proceso de creación de los juegos educativos en eAdventure NO debe ser modificado.

Distribución

Se propone crear nuevas formulas de distribución de juegos educativos sacando provecho a las características tecnológicas y la capacidad de conectividad de los dispositivos móviles. Los objetivos planteados inicialmente en la fase de distribución son los siguientes:

- Crear un sistema de distribución de juegos basada en un repositorio de descarga a modo de tienda de juegos.
- Permitir la distribución directa de juegos entre dispositivos móviles.
- Permitir la distribución típica de juegos entre computadores y dispositivos móviles. (Transferencia por cable).

Ejecución

Como punto más relevante dentro de la realización del proyecto se debe diseñar un nuevo motor de juegos eAdventure capaz de ejecutarse en dispositivos móviles Android. Los objetivos principales definidos en la fase de ejecución son los siguientes:

- Crear un motor de juegos eAdventure para dispositivos móviles Android.

- El motor de juegos debe soportar todas las características de los juegos eAdventure (Juegos de primera y tercera persona).
- El motor de juegos debe permitir la ejecución de los juegos creados anteriormente con la plataforma eAdventure (Compatibilidad hacia atrás)
- El motor de juegos debe permitir una gestión básica de los juegos (Instalación y configuración)
- El motor de juegos debe dar soporte a los objetivos descritos en la fase de distribución.

Desafíos del proyecto

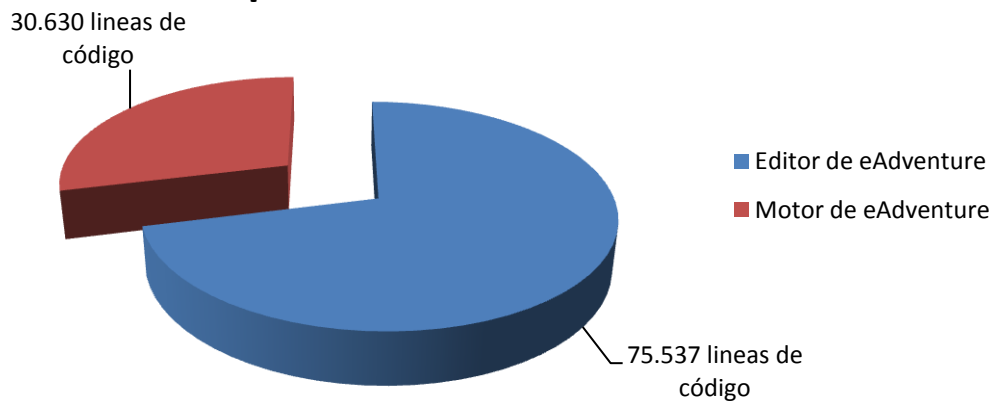
El trabajo a realizar plantea desafíos a la hora de realizar el desarrollo del proyecto.

En primer lugar se debe tener en cuenta que eAdventure es una plataforma de un tamaño considerable. La plataforma ha crecido en los últimos años siguiendo la filosofía de código libre, siendo muchas personas con distintos estilos de programación las que han contribuido en el desarrollo de la plataforma. Además por otro lado eAdventure dispone de poca documentación técnica, siendo esto un reto más para los integrantes del proyecto ya que no tienen conocimiento de la plataforma.

A continuación se muestran una serie de métricas de código relativas al tamaño de la plataforma eAdventure con el fin de aportar un punto de vista de la envergadura de la plataforma.

| Métrica | Valor |
|--|--------------------------|
| Número total de líneas de código de la plataforma eAdventure | 107.432 líneas de código |
| Número total de clases de la plataforma eAdventure | 1304 clases |
| Número total de paquetes | 149 paquetes |

Porcentaje, en líneas de código, de las dos herramientas proporcionadas por la plataforma eAdventure



Por otro lado Android es una plataforma móvil relativamente nueva y en constante crecimiento y evolución. Actualmente se produce una versión nueva de la plataforma Android cada seis meses en la que la plataforma sufre grandes cambios incorporando nuevas funcionalidades. Esto supone un reto para los integrantes del proyecto ya que deben mantener sus conocimientos al día según evoluciona la plataforma, y todo ello sin haber tenido experiencia previa en el desarrollo para Android.

Por otro lado, ha de tenerse en cuenta que el concepto de programación para móviles difiere del concepto de desarrollo típico de aplicaciones de entornos de escritorio. Para desarrollar aplicaciones en móviles se debe conocer bien las tecnologías que incorporan estos dispositivos. Además, en el caso de los videojuegos, los desarrolladores deben comprometerse en mantener código eficiente, ya que éste va a ser ejecutado en dispositivos con limitaciones hardware tanto a nivel de procesamiento como de memoria y batería.

2.4. Metodología y planificación

Para realizar el proyecto se ha decidido planificar el desarrollo del proyecto haciendo uso de la metodología SCRUM. SCRUM es una metodología ágil de desarrollo de software comúnmente utilizado en entornos con un riesgo alto en el desarrollo de productos software.

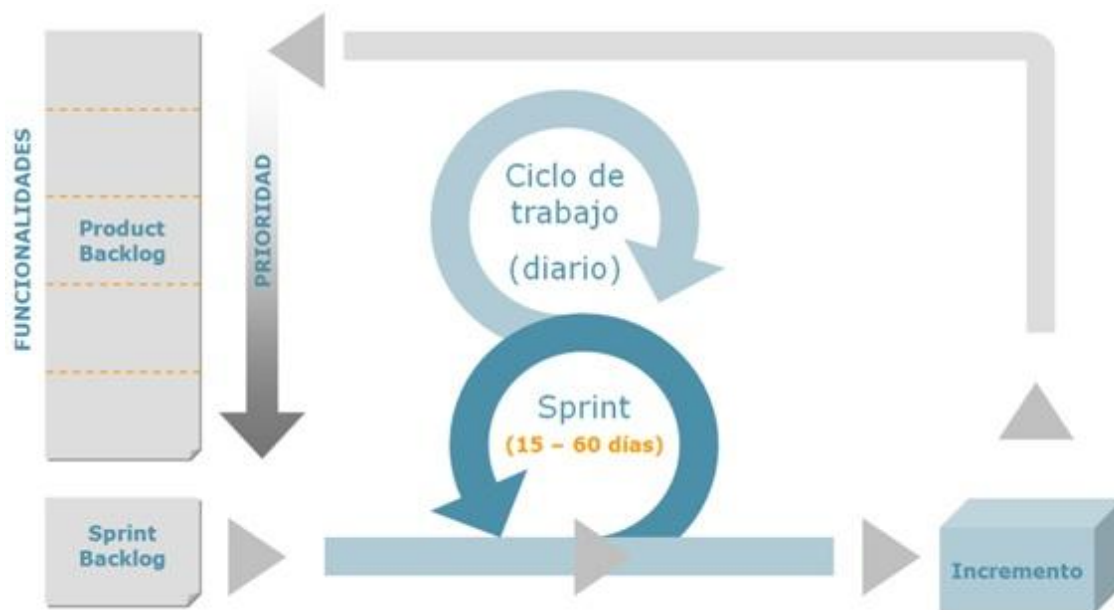


Ilustración 8 - Proceso de desarrollo en SCRUM

El proceso de gestión y desarrollo del software definido en SCRUM ha sido adaptado de manera natural a las necesidades y posibilidades de los integrantes del grupo del proyecto en compatibilidad con sus estudios.

En este caso la planificación se define de la siguiente manera:

- Trabajo diario y reuniones del grupo del proyecto dos veces por semana
- Reuniones semanales formadas por los integrantes del grupo del proyecto y los tutores del proyecto para re-planificar objetivos (*Backlog* de funcionalidades) semanales y revisar los hitos y entregas oficiales.

- Desarrollo incremental del código y entrega de prototipos funcionales
- Presentaciones formales del producto a los directores del proyecto

El proceso permite seguir una gestión regular de las expectativas del proyecto. Siempre se debe tener una lista de requisitos priorizada y actualizada pudiendo realizar una demostración funcional del producto a los tutores del proyecto en cualquier momento. También permite mantener flexibilidad en el desarrollo del producto gracias a la re-planificación realizada en las reuniones semanales y la continua comunicación de los integrantes del grupo.

Para apoyar el proceso se han usado herramientas colaborativas que han ayudado a los integrantes del grupo a mantener una visión global del estado del proyecto en todo momento.

Entre las herramientas se encuentran:

Comunicación y planificación: Google Wave (TaskBoard Widget) y correo electrónico

Desarrollo código: Google Code

Documentación: Google Code Wiki , Google Docs , Microsoft Word

Almacenamiento recursos: DropBox

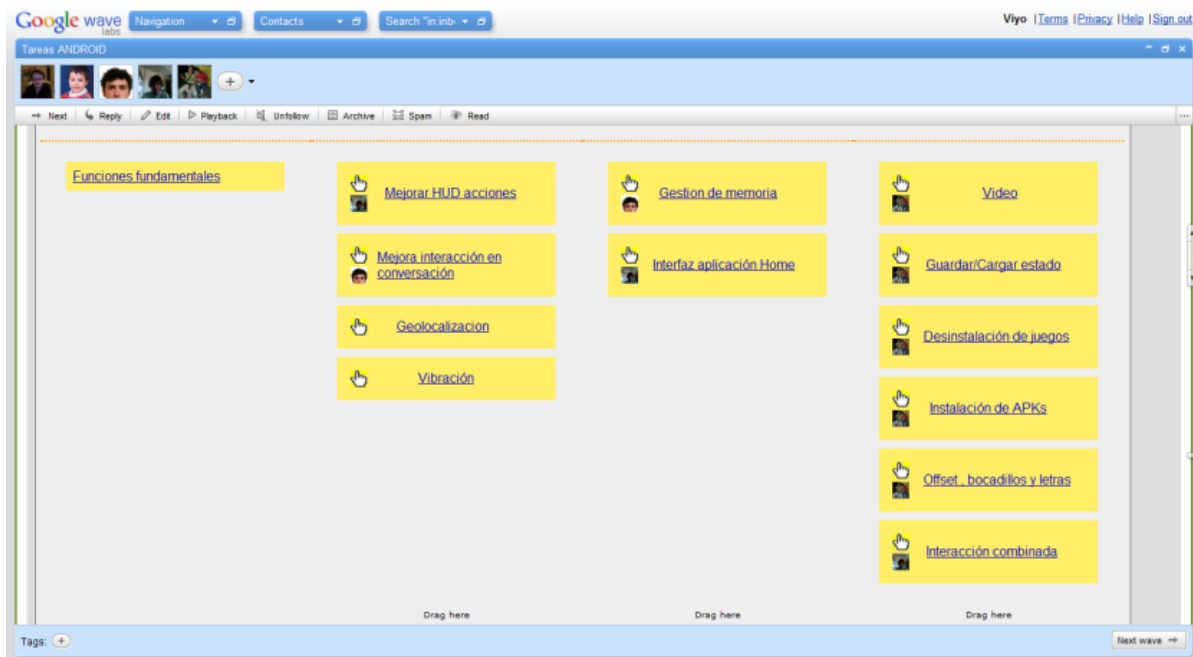


Ilustración 9 - Planificación del proyecto con Google Wave (TaskBoard Widget)

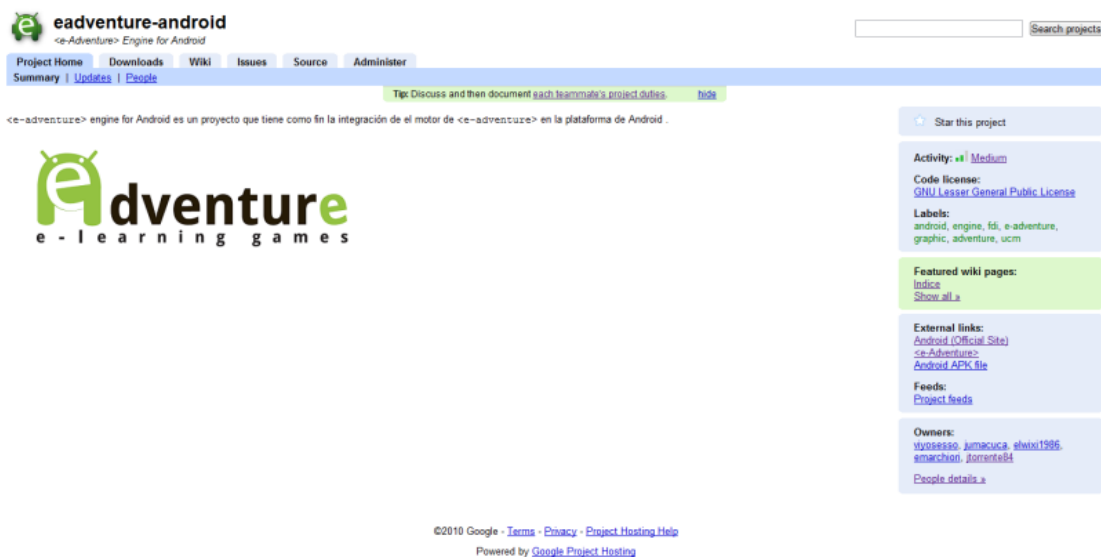


Ilustración 10 - Página Web del proyecto en Google Code

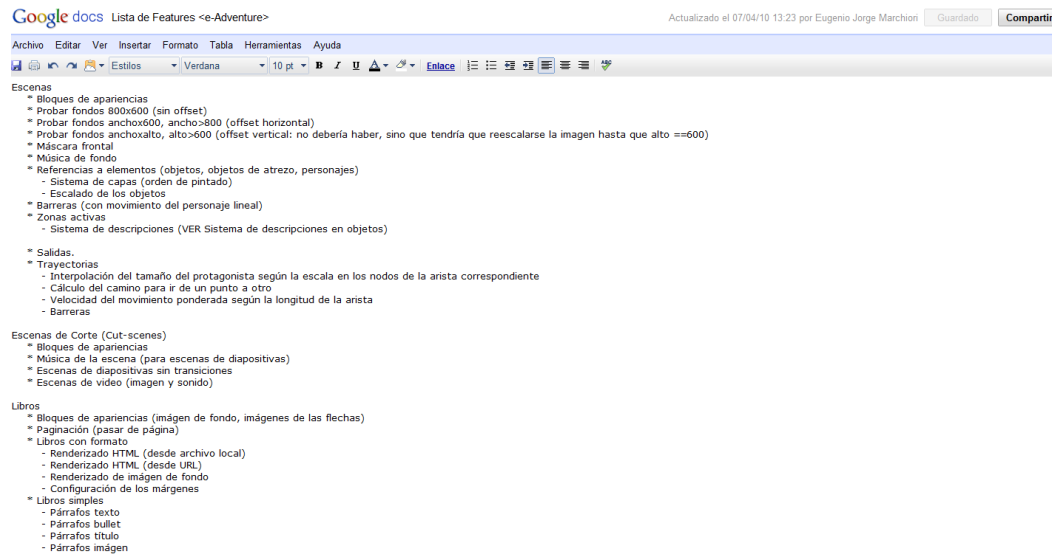


Ilustración 11 - Documentación del proyecto en Google Docs

3. Contexto y trabajo relacionado

El siguiente apartado introduce brevemente el contexto de la temática del proyecto así como el trabajo relacionado.

3.1. M-Learning

En la actualidad cada vez es más accesible la tecnología presente en los ordenadores personales y en los dispositivos móviles permitiendo el acceso a multitud de contenido e información a través de Internet. Existe interés en aprovechar ésta tendencia para propósitos educativos. El término más conocido en éste campo de estudio es “Mobile Learning”:

Mobile Learning se define como la forma de aprendizaje que hace uso de la tecnología en dispositivos móviles, como PDAs, teléfonos móviles y videoconsolas portátiles, para aportar nuevas experiencias educativas en el lugar y momento deseado.

El concepto de Mobile Learning no es nuevo en el ámbito educativo. Mobile Learning se usa en planes de formación interesados en introducir innovaciones tecnológicas. Sin embargo, la mayor parte de estos planes se centran en un modelo de transmisión de información, donde el profesor genera contenidos y los hace accesibles a los alumnos a través del dispositivo, limitándose a la consulta de datos y la interacción guiada a través de respuestas a cuestionarios.

No obstante, existen multitud de áreas educativas en las que m-Learning puede resultar interesantes. En actividades que promueven el aprendizaje a través del refuerzo de la asociación entre un determinado estímulo y una respuesta y actividades en las que los estudiantes utilizan herramientas para crear nuevas ideas y conceptos de forma activa, basándose tanto en sus conocimientos previos como en los adquiridos en el momento.

3.2. Juegos educativos

En las últimas dos décadas los videojuegos han madurado como forma de entretenimiento a nivel mundial. El desarrollo de videojuegos ha evolucionado, en paralelo con la tecnología, dando lugar a distintos tipos de juegos. Más allá de los juegos de acción existen juegos educativos de tipo simulación, estrategia, juegos de rol, deportes, puzzles o aventuras gráficas. Estos juegos pueden ser útiles como medio de aprendizaje ya que enseñan y además potencian distintas habilidades en los estudiantes.

- Manipular variables de otra forma inalterables. (*SimEarth*, el jugador puede observar los efectos de cambiar los niveles de oxígeno de la tierra, o aumentar la temperatura global². Más información en:
- Estudiar fenómenos desde distintas perspectivas. En el juego "*Hidden Agenda*", los jugadores asumen la posición del presidente de un país centroamericano, aprendiendo economía, historia, política, sociología, y cultura durante el proceso³.
- Observar el comportamiento de sistemas a través del tiempo. Por ejemplo, en simulaciones como "*SimCity*" o "*Civilization*", los jugadores pueden observar el comportamiento de sistemas sociales a través de años⁴ o siglos. Donde la mayoría de los modelos físicos tienden a ser estáticos, las simulaciones por ordenador permiten manipular el tiempo.
- Visualizar un sistema en tres dimensiones. En "*Digital Wheater Station*" el aprendiz puede usar herramientas 3D para visualizar sistemas climáticos en tres dimensiones (Hay, K.E. 1999)
- Comprender las diferencias entre un sistema real y una simulación. Las simulaciones no representan la realidad sino la concepción del diseñador de

² <http://www.mobygames.com/game/simearth-the-living-planet>, accedido el 5 de Mayo de 2010.

³ <http://www.mcli.dist.maricopa.edu/proj/sw/games/hidden-agenda.html>, accedido el 7 de Mayo de 2010.

⁴ <http://simcitysocieties.ea.com/index.php>, accedido el 7 de Mayo de 2010.

una realidad (Thiagarajan, 1998). Por ejemplo, en el simulador de ciudades "SimCity" el transporte público tiene un gran peso, reflejo de la afición de su autor por el transporte público (Herz 1997).

Uno de los géneros de videojuegos más versátiles en el campo de la educación es el de aventuras gráficas. En este tipo de juegos, lo más importante es la historia que se cuenta por lo que se puede amoldar de manera eficaz para transmitir conocimientos, o plantear desafíos por resolver. Por otra parte, en las aventuras gráficas la reflexión, el análisis y la deducción a la hora de resolver los problemas priman sobre la acción. Por ello estos juegos son especialmente adecuados para la enseñanza en dominios que requieran el uso de la lógica y el razonamiento.

3.3. Juegos en dispositivos móviles

En los últimos años el número de aplicaciones desarrolladas para dispositivos móviles ha aumentado de forma considerable debido a la continua expansión del mercado de los dispositivos móviles. El total de estas aplicaciones representa un alto porcentaje en el mercado de los videojuegos para dispositivos portátiles afectando directamente a las plataformas más relevantes de videoconsolas portátiles como la *PlayStation Portable* y la *Nintendo DS*.

A continuación se muestra un estudio⁵ donde se estiman los beneficios obtenidos por las plataformas más relevantes en el mundo de los videojuegos para sistemas portátiles en Estados Unidos entre el año 2008 y 2009. Estas plataformas son la *Nintendo DS*, *Playstation Portable* y la plataforma de *iPhone*.

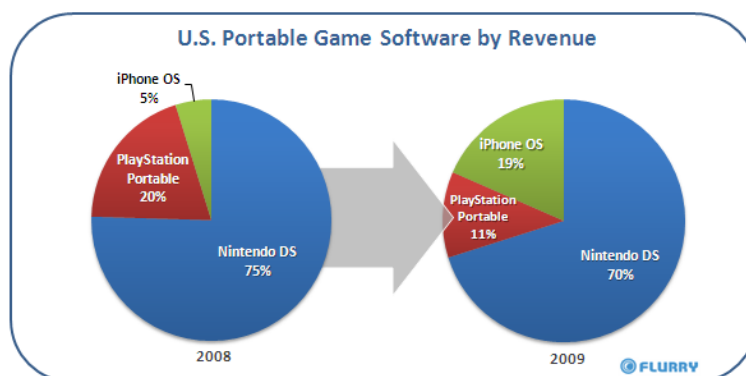
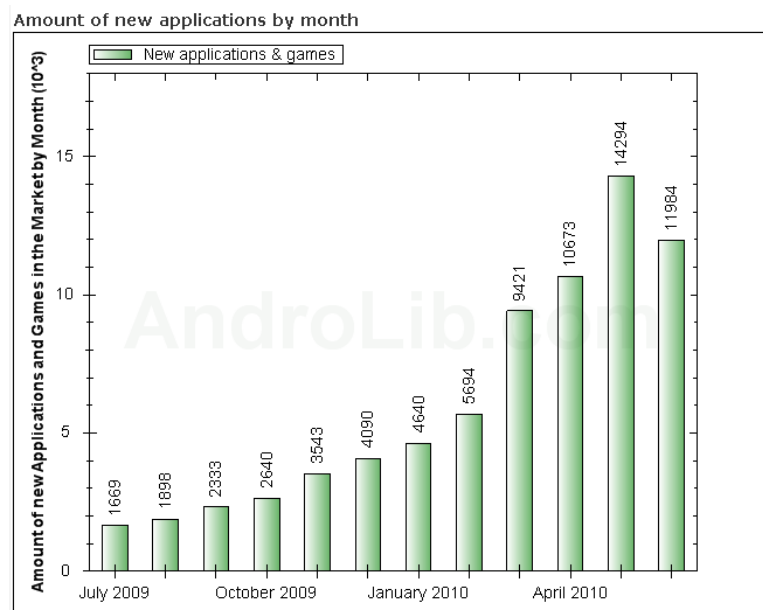


Ilustración 12 - Beneficios del mercado de los videojuegos en Estados Unidos - 2008-2009

Como se observa en el gráfico anterior el uso de las plataformas móviles como plataforma de videojuegos crece considerablemente en los últimos años llegando a competir directamente con las plataformas dedicadas exclusivamente al mundo de los videojuegos.

Por otro lado, como se muestra en la siguiente gráfica en el caso de la plataforma móvil Android, el crecimiento en el desarrollo de videojuegos está en continuo crecimiento.

⁵ Estudio realizado por <http://www.flurry.com>, accedido el 10 de Mayo de 2010



**Ilustración 13 - Cantidad de videojuegos desplegados en la tienda de aplicaciones de Android
Julio 2009 – Junio 2010**

Un estudio realizado por la entidad “Game Developer Research” afirma que el 25% de los juegos desarrollados entre 2009 y 2010 están orientados a dispositivos móviles y en un futuro este porcentaje crecerá.



Ilustración 14 – Relación de los juegos más populares en tres de las plataformas móviles más relevantes. Recolectado por Facebook entre los días 15–29 Enero del 2010

4. Adaptación de juegos educativos a dispositivos móviles

A continuación se realiza un estudio genérico de la adaptación de los juegos educativos a entornos móviles. A lo largo del estudio se toman decisiones de cómo poder integrar las soluciones en la plataforma de eAdventure.

Diseñar aplicaciones para entornos m-Learning no es una tarea trivial. En un principio se pensó que el contenido de un sistema e-Learning podía ser adaptado directamente a un dispositivo móvil. Sin embargo, este enfoque no tiene en cuenta la diferencia de rendimiento que hay entre computadores personales y dispositivos móviles. Por tanto no saca provecho a las características adicionales que estos poseen y no distingue entre las diferencias existentes en el ambiente y filosofía de su uso. Esto conlleva al estudio de cómo y qué factores han de tenerse en cuenta a la hora de diseñar aplicaciones en este tipo de dispositivos (M. Dunlop, 2002).

El proyecto se centrará en proporcionar posibles soluciones de diseño a la adaptación de juegos digitales educativos en entornos m-Learning. Por ejemplo el cambio que supone pasar de trabajar en una computadora de escritorio donde se presenta una cantidad de información abundante, a un dispositivo con tamaño de pantalla reducido y capacidad de entrada limitada, motiva el planteamiento de encontrar un buen diseño de interacción de usuario. Por otro lado, la capacidad de movilidad que tienen los terminales móviles junto con las nuevas tecnologías de conectividad, incitan a diseñar una solución efectiva a la distribución del contenido educativo. Por último se puntúa el uso que se le puede dar a las nuevas tecnologías que se están incorporando en las últimas generaciones de móviles, pudiendo aprovechar esas tecnologías para ofrecer características que en los juegos de escritorio no son viables. La capacidad de los sensores de posicionamiento global, la conectividad a Internet y la utilización elementos hardware de los dispositivos móviles pueden ampliar el valor educativo de los juegos. A continuación realizamos un estudio de los tres puntos anteriores intentando aportar soluciones que sean válidas en cualquier plataforma o dispositivo móvil.

4.1. Interacción de usuario en dispositivos móviles

Es importante diseñar sistemas que sean efectivos, eficientes, sencillos y amenos a la hora de utilizarlos. Más en nuestro caso en el que la interacción de usuario con aplicaciones educativas no debe suponer una barrera en la adquisición de conocimientos.

El diseño de interacción de usuario en dispositivos móviles cuenta con diferencias importantes respecto al diseño típico de interacción en computadores personales. A la hora de desarrollar una solución para dispositivos móviles hay que tener en cuenta factores que lo diferencian de las computadoras personales.

Con el fin de dar una solución eficiente, sencilla e intuitiva buscamos y estudiamos un ejemplo relacionado con la interacción de usuario en un juego de aventura gráfica desarrollada para un dispositivo móvil. Más tarde y basado en las pautas de diseño marcadas por la literatura del campo (J. Gong, 2004) se mostrará una solución posible a la interacción de usuario en aventuras gráficas en dispositivos móviles.

Estudio de Interacción en juegos de tipo aventura gráfica en dispositivos móviles

En este apartado se analiza la interacción de usuario en un juego de tipo aventura gráfica que pertenece a la plataforma *iPhone* prestando especial interés a las siguientes decisiones de diseño:

- Tecnologías de entrada y salida al sistema
- Interacción inicial con la aplicación:
 - Menú inicial
 - Manual de cómo empezar a jugar
- Interacción en el juego

- Distinción de las interacciones en distintos estados del juego
 - Flujo de la interacción

The Secret of Monkey Island Special Edition for iPhone

El juego consiste en una edición especial de la saga *Monkey Island* adaptada para la plataforma *iPhone*. Se escoge esta aventura gráfica por ser una de las más relevantes y difundidas dentro las plataformas analizadas.

La aplicación es una adaptación de la versión de escritorio a la plataforma móvil por lo que resulta interesante estudiarla como ejemplo de adaptación teniendo en cuenta el tamaño reducido de la pantalla, la tecnología táctil, la vibración y los sensores.



**Ilustración 15 –
Splash-Screen del
juego *Monkey Island*
para *iPhone***

Tecnologías empleadas

En primer lugar se analizan las componentes de entrada y salida del sistema que permiten a los jugadores interactuar con el juego.

Como entrada al sistema se hace uso de:

- Gestos sobre la pantalla táctil de uno o varios dedos.
- Sensor de acelerómetro.

Como salida se hace uso de

- Pantalla
- Audio
- Vibración

Estas tecnologías están ampliamente expandidas en los dispositivos móviles actuales con lo que en principio resulta familiar su uso para cualquier usuario dueño de este tipo de dispositivos.

Menú inicial

Al iniciar la aplicación se muestra un menú desde el cual se puede optar por iniciar una partida nueva, cargar o continuar una partida anterior o acceder a un apartado de ayuda y opciones.

Para acceder a cada sección el usuario deberá pulsar sobre una de las opciones para, en primer lugar, seleccionarla y volver a pulsar para finalmente acceder a ella.

Esta decisión llama la atención ya que parece que tiene que ver con asegurarse de la precisión en la pulsación y así evitar errores de selección en pantallas de tamaño reducido. En cualquier caso observando juegos pertenecientes a la misma plataforma se ve que en general no se usa este tipo de interacción y la precisión táctil es óptima.

Carga de partida / Continuar partida

En el caso de que el usuario acceda a cargar o continuar una partida, aparecerá una tabla con la lista de las últimas partidas guardadas. A primera vista se muestran las tres últimas partidas guardadas (si existen), pero en caso de querer acceder a partidas anteriores se dispone de dos flechas con las que navegar por el panel. El tamaño de las celdas es suficientemente grande para presionarla con el dedo y así facilitar su selección. Aun así la selección se ha implementado de la misma manera que en el caso anterior

y se necesitara pulsar dos veces para poder iniciar la partida



Ilustración 16 - Menú de carga de partidas del juego de Monkey Island para iPhone

guardada. La información que se muestra al usuario referente a las partidas guardadas es la siguiente:

- Tipo de guardado: automático o manual
- Nombre descriptivo de la parte del juego en la que se produjo el guardado
- Fecha y hora del guardado
- Tiempo de juego desde que se inicio la partida
- Porcentaje completado del juego

Ayuda y Opciones

Dentro del apartado de ayuda y opciones el usuario puede consultar y configurar distintas opciones que van a influir directamente en la interacción con el juego.

En los apartados de “cómo se juega” y “controles” se puede consultar como ha de realizarse la interacción con la dispositivo para jugar a la aventura, todo esto en una serie de diapositivas explicativas. Además en el apartado de subtítulos se puede configurar para que se muestren los diálogos como texto a la vez que son reproducidos por audio.



Ilustración 17 - Tutorial de interacción en Monkey Island para iPhone

Un punto que resulta interesante es el que el usuario dispone de un manual de cómo se juega a la aventura gráfica, en el que siempre que quiera puede acceder y consultarlo.

Interacción en el juego

Para analizar la interacción en el juego se hace una división de los principales estados del juego y se analiza la interacción por separado en cada uno de ellos.

Estos son:

- Escenas
- Escenas de corte (videos o *cutscenes*)
- Conversaciones
- Tiempos de carga de juego
- Menú de pausa

Escenas

Una vez iniciada la partida, las escenas se representan en horizontal o modo apaisado. El contenido ocupa toda la pantalla pudiendo ser el tamaño de la escena más ancha pero siempre igual de alta que la pantalla. El personaje al desplazarse y llegar a las zonas



Ilustración 18 - Escena de juego en Monkey Island para iPhone

límites de izquierda o derecha hace cambiar la cámara de fondo de la escena permitiendo que se vea la parte que le acompaña. En cambio la cámara nunca cambiará cuando el personaje se encuentre en los límites superior e inferior ya que la escena tiene exactamente el alto de la pantalla del dispositivo.

Puntero

Para que el jugador interaccione con la escena se ha optado por el uso de un puntero, al igual que en el juego de escritorio. El puntero se desplaza por la escena cuando el



Ilustración 19 - Animación de puntero en la interacción. Monkey Island para iPhone

usuario hace gestos de presión y desplazamiento en cualquier punto de la pantalla con el dedo. De manera que:

“Estando el puntero en una posición X , el usuario puede presionar y desplazar en cualquier punto de la pantalla Y un recorrido R que moverá el puntero desde su posición inicial X la distancia R en la misma dirección.”

Por otro lado el puntero lleva asociado un icono que representa la acción seleccionada en ese momento. Cuando el jugador interaccione con un elemento de la escena debe

colocar el puntero sobre el elemento de la escena y posteriormente realizar una pulsación en cualquier punto de la pantalla para que se ejecute la acción seleccionada sobre el elemento al cual apunta el puntero. Esto es la acción análoga a hacer “click” en el juego de escritorio sobre el elemento deseado.

Éste tipo de interacción no resulta ser muy intuitivo. Actualmente no hay ningún dispositivo móvil, sin tener en cuenta netbooks o tablet pcs, que disponga de puntero para interaccionar con el sistema. En el caso de los tablet PCs el cursor no se desplaza sino que cuando el usuario pulsa una zona de la pantalla este reaparece en ese punto y el sistema interpreta un “clic” automáticamente.

Head-Up Display

Normalmente los videojuegos cuentan con información que se muestra continuamente a lo largo del juego. A este contenido se le denomina HUD (*head-up display*) y en este caso se representa mediante una barra situada en la parte inferior de la pantalla. El HUD cuenta con dos botones que activan los paneles del menú de acciones del personaje y el menú de inventario. Además cuenta con información textual de la acción seleccionada en todo momento.

Acciones

Para acceder al menú de acciones el usuario debe presionar el botón asociado en la zona



Ilustración 20 - Menú de acciones. Monkey Island para iPhone

izquierda de la barra inferior del HUD. Para cerrarlo basta con volver a presionar el mismo botón.

Para seleccionar una acción el usuario debe pulsar sobre el icono que la representa teniendo las siguientes consecuencias a nivel visual:

- El botón del menú de acciones cambia a uno que hace referencia a la acción.
- El texto asociado al botón del menú de acciones informa de la acción seleccionada
- El icono asociado al cursor cambia a otro que hace referencia a la acción.

Estos tres efectos ayudan al usuario a recordar la acción seleccionada en cada momento.



Ilustración 21 - Acción asociada al puntero. Monkey Island para iPhone

Inventario

Para acceder al inventario se debe presionar el botón correspondiente situado a la derecha en la barra inferior del HUD. Para cerrarlo bastará con presionar de nuevo el



Ilustración 22 - Menú de inventario. Monkey Island para iPhone

mismo botón. El inventario se representa con una matriz de celdas que contienen elementos recogidos por el protagonista a lo largo del juego. El número de elementos a priori es ilimitado con lo que el panel dispone de dos flechas para poder navegar a través de ellos. La navegación se realiza con pulsaciones sobre las flechas de desplazamiento correspondientes.

Flujos de interacción en las escenas

Mover al personaje

Para desplazar al personaje por la escena el jugador tan solo debe asegurarse de que está seleccionada la acción predeterminada de “mover personaje” y después

- Situar el cursor donde quiera que vaya
- Hacer una pulsación sobre cualquier punto de la pantalla para que se ejecute la acción

Consultar inventario

Para consultar el inventario el jugador tan solo debe presionar sobre el icono del menú de inventario.

Realizar acciones

Después de haber seleccionado una acción el usuario puede aplicarla moviendo el cursor sobre cualquier elemento de la escena. Para informar visualmente al usuario de que el cursor está situado sobre un elemento interactivo, el cursor se anima junto con su icono. Además estando el cursor sobre un elemento de la escena se muestra su acción predeterminada de forma textual y en la parte superior izquierda de la pantalla. Para terminar de realizar la acción sobre el elemento basta con hacer una pulsación en cualquier punto de la pantalla.

En el caso de querer realizar una acción sobre algún elemento del inventario, primero se debe seleccionar la acción y posteriormente acceder al menú de inventario y presionar sobre la celda que contiene el elemento deseado. Si se quiere realizar alguna acción combinada con algún elemento del inventario, se debe seleccionar en primer lugar la acción, a continuación seleccionar el elemento del inventario, mover el cursor sobre un elemento de la escena y presionar cualquier zona de la pantalla. También se puede combinar con otro elemento del inventario. Para ello basta con seleccionar la acción, el primer objeto del inventario y por último el segundo objeto del inventario.

Escenas de corte

En este caso se puede decir que las escenas de corte son secuencias no interactivas que se usan tan solo para mostrar información del capítulo aprovechando la carga del siguiente capítulo. Destaca el hecho de que aprovecha el espacio completo de la pantalla y que el paso al siguiente capítulo es automático y no requiere de interacción por parte del usuario.



Ilustración 23 - Escena de corte. Monkey Island para iPhone

Conversaciones

Las conversaciones en el juego se muestran a pantalla completa pero con la barra inferior del HUD oculta ya que no es de utilidad en este estado del juego. La región que

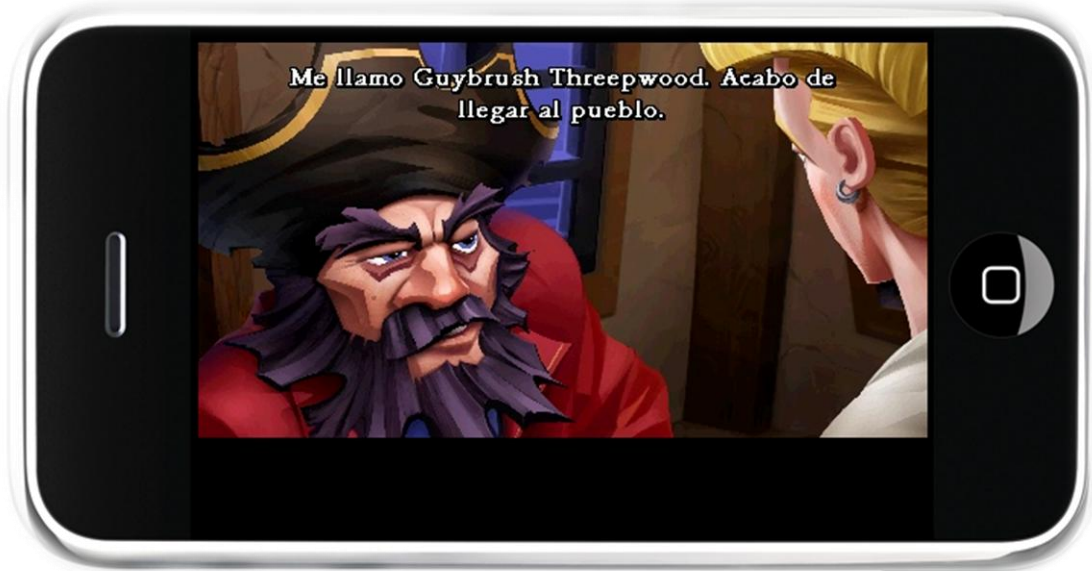


Ilustración 24 - Conversación en Monkey Island para iPhone

ocupaba la barra simplemente se colorea en color negro.

De esta manera no aprovechan el espacio completo de la pantalla siendo algo que debe tenerse en cuenta a la hora de desarrollar contenido para este tipo de dispositivos.

Los diálogos se representan textualmente con letras de distintos colores asociadas a cada personaje. Las letras tienen un borde para resaltar sobre el fondo de la escena y así facilitar la visibilidad al usuario. Las frases de diálogo están dispuestas teniendo en cuenta la posición del personaje, situándose junto a él. Esto ayuda a identificar fácilmente el personaje emisor en cada momento de una conversación.



Hay partes del diálogo en las que el jugador debe escoger entre una serie de opciones para poder continuar con la conversación. El usuario debe presionar sobre una de ella para seleccionarla y volver a presionar para validar la elección.

Además los diálogos son reproducidos por audio, siendo una opción del juego el activar o desactivar los textos de las conversaciones de manera que tan solo se escuchen los diálogos.

Carga del juego

Hay estados del juego donde el usuario debe esperar a que se cargue el contenido para poder continuar con la partida. En estos estados se muestra a pantalla completa una



Ilustración 25 - Carga de recursos en Monkey Island para iPhone

animación del protagonista sin información textual que indica la carga de contenido.

Menú pausa del juego

El juego cuenta con un menú de pausa para cuando el jugador quiere terminar la partida, guardar el estado o configurar algunos parámetros del juego. Para ello han utilizado la capacidad del *iPhone* de controlar la orientación del dispositivo a través del sensor de acelerómetro, de manera que cuando se pone el dispositivo en orientación vertical se muestra el menú. Esta decisión nos ha parecido interesante al no haber interferido de ninguna manera en la interfaz principal del juego.



Ilustración 26 - Menú de pausa del juego Monkey Island para iPhone

Posible solución a la interacción en juegos de tipo aventura gráfica en dispositivos móviles

Basado en las reflexiones que se han hecho tras el análisis de la aventura gráfica *Monkey Island* para *iPhone*, se estudia y diseña una posible solución genérica, aplicable a eAdventure, teniendo en cuenta los siguientes aspectos:

- Compatibilidad con distintas características de los dispositivos móviles
- Aprovechamiento del espacio de la pantalla
- Interacción sencilla e intuitiva

Con ello, la solución debe mantener un alto grado de compatibilidad con los dispositivos móviles actuales. Además debe ser válida y reutilizable en la mayoría de dispositivos del mercado, con lo que al diseñarlo se tendrá en cuenta el uso de las tecnologías más expandidas en dispositivos móviles modernos. A su vez debe ser intuitiva y sencilla de aprender facilitando la interacción entre persona y dispositivo móvil.

Tecnologías “Input” y “Output”

A continuación se muestran las tecnologías de entrada y salida al sistema escogidas para la interacción:

- Como entrada al sistema - *input* :
 - Gestos táctiles de un solo dedo sobre la pantalla del dispositivo móvil
 - Botón físico
- Como salida del sistema - *output* :
 - Pantalla táctil
 - Audio
 - Vibración

Visualización del juego

En primer lugar se toma la decisión de cómo debe ser visualizada la aventura gráfica en un dispositivo con una pantalla de tamaño reducido. Siempre se busca optimizar el espacio de la pantalla con lo que se decide representar la aventura gráfica en modo apaisado u horizontal, ocupando así el mayor espacio disponible como se muestra en las siguiente imagen.



Ilustración 27 - Visualización de una escena en eAdventure para Android

Tiene sentido el que el contenido de las aventuras gráficas, por ejemplo las escenas y los videos, sean más anchos que altos adaptándose mejor a esta configuración de pantalla. Además las consolas portátiles actuales cuentan con una configuración de pantalla apaisada y panorámica con lo que se sigue la línea marcada por la industria repercutiendo en una interacción más intuitiva y agradable de usar.

Interacción en el juego

La segunda decisión que se ha tomado es diferenciar los componentes comunes de los juegos de tipo aventura gráfica para definir la interacción por separado en cada uno de ellos.

Detectamos cuatro componentes comunes de los cuales se define la interacción en cada uno de ellos por separado. Estos son las escenas, escenas de corte (video y diapositivas), conversaciones y libros.

Escenas

Las escenas son los componentes principales de las aventuras gráficas. Es donde el jugador interacciona con el entorno del juego. Son las más complejas, en cuanto a interacción ya que es donde el usuario puede hacer la mayoría de acciones.

Movimiento del personaje

En las aventuras gráficas de tercera persona el jugador “controla” al protagonista dentro del juego, con lo que se debe definir la manera de interactuar para desplazar al protagonista por la escena de juego. Siempre se ha intentado que la interacción sea lo más intuitiva posible, con lo que se ha tenido en cuenta el gesto análogo a la interacción que la mayoría por no decir todas las aventuras gráficas del mercado usan para mover al personaje en un entorno de escritorio. Ésta interacción es hacer “click” con el puntero. El gesto análogo a hacer “click” en una pantalla táctil es hacer una pulsación simple sobre esta, siendo en los dispositivos móviles táctiles la interacción principal para seleccionar los elementos con los que interactuar. Por lo tanto parece lo más intuitivo el usar este tipo de interacción para desplazar al personaje.



Ilustración 28 - Movimiento del protagonista en eAdventure para Android

Además se ha decidido diseñar una respuesta visual a la interacción del movimiento del personaje. Como se puede observar en la ilustración anterior se ha diseñado una onda de color azul en la zona de pulsación de manera que el usuario sabe en todo momento en que dirección se debe desplazar el personaje.

Consulta del inventario

Para consultar el inventario, el usuario debe hacer un barrido vertical de arriba a abajo mientras se encuentra jugando en una escena para arrastrar el inventario sobre ésta.

El inventario se representa mediante una rejilla traslúcida por la cual se puede navegar para consultar los distintos elementos almacenados como se puede observar en la Ilustración 29.

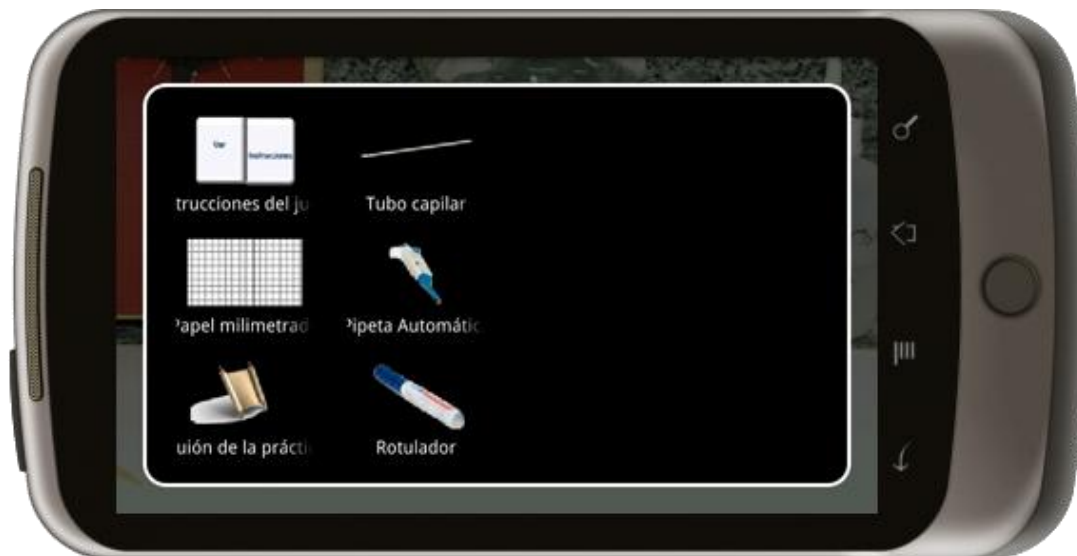


Ilustración 29 - Menú de inventario en eAdventure para Android

Para navegar por el inventario el usuario debe arrastrar el inventario presionando con el dedo. Para seleccionar un elemento del inventario basta con hacer una pulsación simple sobre el objeto deseado.

Para cerrar el inventario, el usuario debe hacer la acción contraria a abrir el inventario, debe realizar un barrido vertical de abajo a arriba de la pantalla del inventario para que este desaparezca y aparezca la escena del juego.

Interacción con elementos

Para interactuar con los elementos de la escena el usuario debe primero realizar una “búsqueda del elemento” y más tarde seleccionar la acción a realizar.

Un elemento con el que se puede interactuar puede estar situado en distintos lugares:

- Elemento del inventario: el usuario debe “Consultar el inventario” y navegar por los elementos hasta que se encuentre el elemento deseado. Al pulsar en él se muestra el menú de acciones asociado a ese elemento.
- Elemento situado en la escena: el usuario debe pulsar la pantalla con un dedo e ir arrastrando este por la escena en busca de elementos. A este gesto se le denomina “Búsqueda de elementos”

Búsqueda de elementos

La acción de buscar elementos viene acompañada de una lupa situada bajo el dedo que se arrastra por la pantalla. La lupa muestra una imagen ampliada de lo que se encuentra bajo el dedo que realiza la interacción. Al encontrar un objeto interactivo bajo la lupa, la lupa cambia de color, se muestra el nombre del elemento interactivo y el dispositivo realiza una vibración con el fin de informar al usuario de distintas formas.



Ilustración 30 - "Búsqueda de elementos" en eAdventure para Android

Cuando el usuario levanta el dedo tras haber estado situado sobre un elemento interactivo de la escena, se muestra automáticamente el menú de acciones asociado a ese elemento. Esta consecución de interacciones, es el análogo a mover el cursor por la escena en busca de los elementos interactivos y hacer "click" en alguno de ellos para mostrar sus acciones asociadas.

Salidas

Las salidas también se encuentran mientras se realiza la búsqueda de elementos. En el caso de que se encuentre una salida, la lupa cambia de color y el dispositivo vibra para que el usuario perciba que es una salida y al soltar el dedo se ejecuta la acción asociada a la salida como se muestra en la siguiente figura.



Ilustración 31 - Interacción en salida en eAdventure para Android

Menú de acciones

El menú de acciones es similar al menú del inventario. Contiene las acciones asociadas al elemento previamente seleccionado. De la misma manera que el del inventario, si contiene más de tres acciones se puede navegar por el menú realizando un arrastre de izquierda a derecha. En la parte superior izquierda del menú, se muestra el icono del elemento seleccionado y su nombre como se puede apreciar en siguiente figura.



Ilustración 32 - Menú de acciones en eAdventure para Android

Para seleccionar una acción basta con presionar en el botón de la acción deseado. Al presionar el botón el dispositivo móvil vibra y el botón asociado a la acción cambia de color para dar una respuesta visual de la interacción al usuario.

Acciones de dos pasos

Hay elementos que pueden ser combinados con otros elementos, a estas acciones combinadas se las llama acciones de dos pasos. Para ejecutar estas acciones se sigue un procedimiento igual que el anterior pero con un paso adicional. Se comienza haciendo una “búsqueda de elementos”, tras encontrar y seleccionar el objeto deseado se ejecuta la acción de dos pasos deseada que se muestra en el menú de acciones, y tras haber seleccionado esa acción se busca el elemento con el que combinar el anterior objeto. La siguiente ilustración muestra la interacción en acciones de dos pasos.



Ilustración 33 - Acción combinada en menú de acciones en eAdventure para Android

La búsqueda del segundo elemento con el que interactuar debe ser relativamente rápida debido que si al cabo de un tiempo no se encuentra el elemento con el que hacer la acción de dos pasos, el reconocimiento de esa acción es desestimado. Como viene haciéndose en la búsqueda de objetos, para dar facilidad al usuario para el reconocimiento de los objetos, si durante la búsqueda se pasa por el objeto con el que

hacer la acción de dos pasos, el color de la lupa cambia y el dispositivo móvil vibra, además sobre la lupa se muestran los nombres de los dos elementos que realizan la acción de dos pasos como se muestra en la siguiente ilustración.



Ilustración 34 - Interacción combinada en eAdventure para Android

Libros

Los objetos libro se muestran a pantalla completa. Para pasar las páginas del libro basta con hacer una interacción sencilla de pulsación simple.

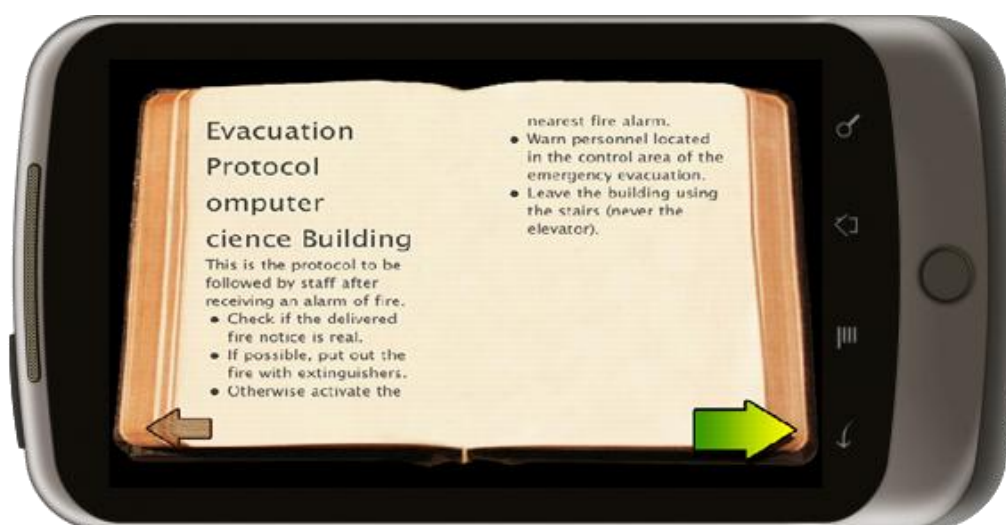


Ilustración 35 - Libro en eAdventure para Android

Escenas de corte

- Escenas de diapositivas: Este tipo de escenas carecen de elementos con los que interactuar. La interacción en este tipo de escenas es sencilla, el usuario debe realizar una pulsación simple sobre cualquier punto de la pantalla para pasar las diapositivas.



Ilustración 36 - Escena de diapositivas en eAdventure para Android

- Escenas de video: En este tipo de escenas al igual que las escenas de diapositivas carece de elementos interactivos. La interacción en este tipo de escenas es también sencilla, el usuario debe realizar una pulsación simple sobre cualquier punto de la pantalla para finalizar la reproducción del video.

Diálogos

Nodos de conversación

Las conversaciones deben ser representadas junto al personaje que habla en cada momento. Si en el dialogo existe un nodo con una frase de un personaje demasiado larga para ser dibujada en una línea, esta se divide en varios nodos, que pasan a una velocidad determinada para facilitar la lectura a los usuarios.



Ilustración 37 - Conversación en eAdventure para Android

Si el usuario desea pasar por alto una conversación debe hacer una pulsación simple en cualquier zona de la pantalla.

Nodos de opciones

Los nodos de opciones son los nodos en los que el usuario puede introducir respuestas en el juego para continuar en la conversación. Las opciones se muestran en forma de lista deslizable con un tamaño de letra razonable para poder ser leída y pulsada. Para seleccionar una opción existen dos posibilidades:

- Se puede hacer una pulsación simple sobre una de las opciones.
- Se puede utilizar el *trackball* para moverse a través de las distintas opciones y seleccionar una de ellas presionando el botón.

Si la línea de las opciones es demasiado larga, esta se anima de derecha a izquierda de manera que se puede visualizar la frase completa.

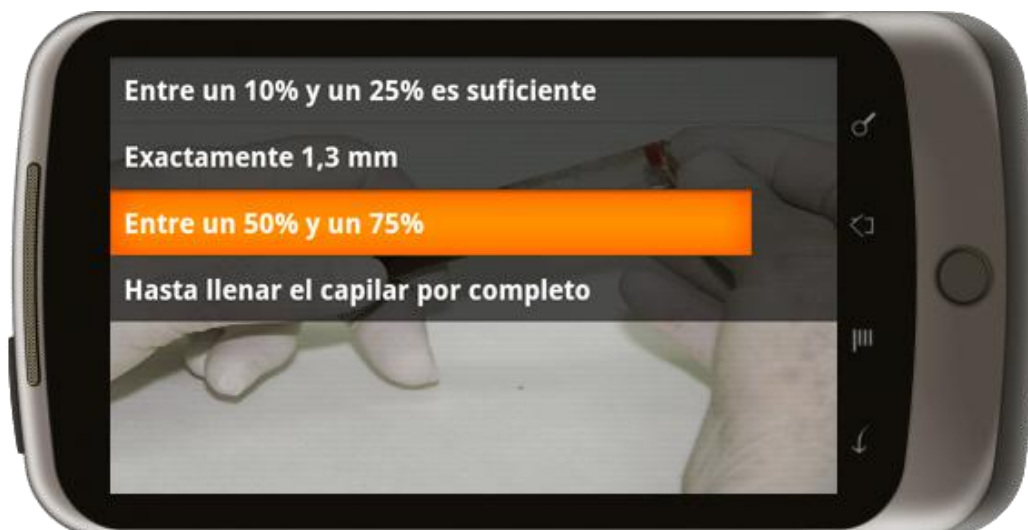


Ilustración 38 - Opciones en conversación en eAdventure para Android

Menú del juego

Para mostrar el menú del juego se debe pulsar el botón físico menú del dispositivo móvil.



Ilustración 39 - Menú de juego en eAdventure para Android

Para interactuar con el menú basta con pulsar el botón asociado a la acción deseada.

Interacción en aventuras de primera persona

Se diferencia la interacción en las aventura gráficas de primera con las de tercera persona en la ausencia del movimiento del personaje. Por ello la interacción asociada al movimiento del personaje no tiene sentido en este tipo de juegos. De esta manera cuando el usuario presiona sobre la pantalla se muestra una “onda” azul animada para indicar la detección de la pulsación, pero no tendrá ningún efecto sobre la escena.



Ilustración 40 - Aventura gráfica en primera persona en eAdventure para Android

4.2. Distribución de juegos educativos en entornos m-Learning

En el siguiente apartado se plantean distintas formas de distribuir contenido educativo, en concreto videojuegos educativos, en entornos m-Learning.

Conectividad en dispositivos móviles

Actualmente los dispositivos modernos cuentan con tecnologías que dotan al terminal de múltiples opciones de conectividad. La opción de conectarse a Internet a través de tecnologías inalámbricas permiten que se ideen nuevas fórmulas que complementan la distribución típica de almacenamiento e instalación de videojuegos a través de medios físicos digitales como puede ser una tarjeta de memoria externa SD. A pesar de ello es común el que dentro de las plataformas móviles existan ciertas restricciones a la hora de distribuir aplicaciones ejecutables dentro de ella. Por ejemplo en el caso de la plataforma *iPhone*, la distribución de los juegos se ve afectada por su política de distribución de aplicaciones y todo debe distribuirse a través de su tienda de aplicaciones *on-line*. En la reflexión que se realiza a continuación no se tienen en cuenta las restricciones propias de cada plataforma. Se tratará de proporcionar soluciones generales que puedan ser aplicadas en la mayoría de ellas.

En primer lugar se estudian los distintos medios de transferencia que pueden ser utilizados para la distribución de los juegos.

Transferencia directa desde el PC

La primera solución posible es transferir los juegos desde el propio ordenador al dispositivo móvil. De esta manera basta con conectar el dispositivo móvil al ordenador, transferir el juego a la memoria interna o externa del dispositivo e instalarlo con una

aplicación de instalación proporcionada por la plataforma. El proceso puede variar dependiendo de cada plataforma pero debería ser un proceso similar en todas ellas.

Este tipo de distribución puede ser útil, por ejemplo, en el caso de un desarrollador de juegos que requiere de un despliegue inmediato de los juegos en desarrollo para hacer pruebas.



Ilustración 41 - Ejemplo de transferencia directa USB

Transferencia directa entre dispositivos móviles

Otra solución posible es la transferencia de los juegos entre dispositivos móviles utilizando tecnologías de conectividad de campo cercano como Bluetooth o infrarrojos. De esta manera los usuarios de los dispositivos se ponen de acuerdo en la transferencia de los juegos y no necesitan un ordenador personal como medio de transmisión.

Este tipo de distribución es interesante en contextos donde no se dispone de un computador personal y la transferencia se hace directamente entre terminales móviles.



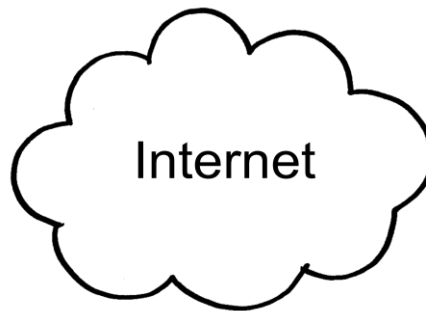
Ilustración 42 - Ejemplo de transferencia de campo de cercano Wi-fi y Bluetooth

Distribución a través de Internet

La distribución a través de Internet es la opción más flexible de todas ellas. Actualmente la mayoría de dispositivos móviles cuentan con acceso a Internet permitiendo en cualquier momento el acceso a la descarga de contenido. Dentro de este punto se analizan dos posibles soluciones:

Basado en repositorio

Dentro del campo de distribución a través Internet, el sistema más común es la distribución abierta de contenidos a todos los usuarios interesados. Para esta opción es necesario un servidor en el cuál están los contenidos que se distribuyen. Es una solución muy versátil puesto que esos contenidos son accesibles a cualquier usuario siempre que disponga de conectividad a Internet



Integrado dentro de un sistema de gestión e-Learning

En este caso el sistema de distribución puede estar integrado dentro de un sistema gestor de contenidos educativos (LMS) utilizado por el centro, o institución educativa. En este caso se dan ventajas desde el punto de vista educativo, ya que por ejemplo permite la adaptación de los juegos a los alumnos de forma individualizada, teniendo en cuenta la capacidad y características de cada uno de ellos. Los juegos quedan integrados en el flujo de las asignaturas virtuales y permite al docente la gestión de estos a través de la aplicación dedicada a la enseñanza electrónica

4.3. Nuevas tecnologías en dispositivos móviles, GPS y cámara de fotos

A continuación se estudian nuevas formas de interactuar en aventuras gráficas a través del uso de las tecnologías disponibles en los dispositivos móviles de última generación, como el posicionamiento global (GPS) y el uso de la cámara, permitiendo nuevas formas de juego que aumentan el valor educativo de estos. Este estudio permitirá más adelante decidir como incorporar estas tecnologías en el flujo de los juegos de eAdventure.

Estudio de las tecnologías

Gracias a la movilidad que ofrecen este tipo de dispositivos es posible alejarse del escritorio y salir al mundo real manteniendo la capacidad de realizar las tareas típicas de los ordenadores tradicionales. Por lo tanto se plantea un reto en el paradigma de los juegos educativos para alcanzar mejores vínculos entre la realidad virtual y el entorno real del jugador. Esto supone una mejora importante para el desarrollo de experiencias educativas en el lugar y el momento deseado “Just In Time Learning”.

Tecnologías disponibles

A continuación se describen dos tecnologías disponibles que permiten aumentar el valor educativo de los juegos.

Cámara de fotos

Prácticamente todos los dispositivos móviles de última generación incorporan un sensor óptico que permite capturar fotografías o grabaciones de vídeo. Éste componente permite interactuar directamente con el entorno real obteniendo información a partir de

él a través de la toma de imágenes y el uso de software de reconocimiento de patrones y objetos.

Sistema de geolocalización

Otra tecnología cada vez más extendida en dispositivos móviles es la recepción de señal de posicionamiento global (GPS), que permite conocer las coordenadas geográficas del usuario en el momento del juego, es decir, el lugar en el que se encuentra en el mundo real. Ésta herramienta es útil para cualquier juego enmarcado en un lugar determinado, permitiendo nuevos modos de interacción basados en la posición real del usuario.

Uso de las nuevas tecnologías

A continuación se analiza el uso de este tipo de tecnologías en juegos basados en aventura gráfica.

Juegos geolocalizados

En el caso de los juegos de tipo aventura gráfica existen varias opciones para integrar ésta tecnología en el flujo de los juegos.



Por ejemplo, la transición entre escenas de un juego puede estar condicionada por la localización del jugador de forma que el juego cambie a una escena diferente en función del lugar. Éste comportamiento permite la creación de juegos que actúen a modo de guía durante una visita turística permitiendo, por ejemplo, mostrar contenidos multimedia con información complementaria sobre los elementos que rodean al jugador.

Otro ejemplo es el caso de los juegos de tipo *gymkhana* en los que el jugador debe desplazarse a otro lugar para superar etapas del juego. Infinidad de juegos o actividades que impliquen el movimiento del jugador en un entorno real y al aire libre pueden beneficiarse de esta tecnología.

Juegos con códigos bidimensionales

Los códigos bidimensionales son imágenes que contienen información codificada en forma de matriz de puntos. Utilizando la cámara de fotos es sencillo decodificar la información que contienen dichos códigos a través de distintas técnicas y algoritmos de reconocimiento de patrones.



Ilustración 43 - Código QR correspondiente a la página Web del proyecto

Existen multitud de juegos en los que esta tecnología es aplicable como medio de interacción con elementos del mundo real. Imagínese un juego diseñado para complementar una visita a un museo, donde cada muestra del museo cuenta con un código bidimensional asociado. El jugador podrá acercarse a la muestra y capturar con la cámara de fotos su código asociado, teniendo el efecto, por ejemplo, de reproducir un video relacionado con la muestra.

Con ello, los códigos bidimensionales son adecuados en situaciones interiores o espacios cerrados donde la localización por sistemas GPS se limita a la recepción de la señal. Sin embargo conlleva una tarea adicional para el desarrollador ya que debe colocar los códigos impresos en los distintos lugares relacionados con el juego.

5. Análisis de las plataformas escogidas

A continuación se analizan con detalle las plataformas en las que se basa el desarrollo del proyecto.

5.1. Análisis de eAdventure

En el siguiente apartado se analiza la plataforma eAdventure. El análisis comienza con la descripción del tipo de juegos soportado por la plataforma. Más adelante se describen las herramientas proporcionadas por la plataforma para la creación y ejecución de los juegos. Por último se realiza un acercamiento a las tecnologías utilizadas en el desarrollo de la plataforma

Aventuras gráficas

La plataforma eAdventure permite la creación de juegos de tipo aventura gráfica. La dinámica de este tipo de juegos consiste en avanzar por el mismo resolviendo diversos rompecabezas, planteados como situaciones que se suceden en la historia, interactuando con personajes y objetos, moviendo al personaje y realizando distintas acciones sobre elementos de las escenas.

Los juegos de la plataforma eAdventure se pueden clasificar dentro de aventuras gráficas de primera y tercera persona :

- En las aventuras gráficas de tercera persona se incluye una visión del personaje principal controlado por el usuario que realiza las acciones indicadas.

- En las aventuras de primera persona no existe ningún protagonista principal sino que el juego se muestra de forma que el jugador es el protagonista y ve la escena como si estuviera presente en ella.



Ilustración 44 - Juego de tipo aventura gráfica de eAdventure

Juegos eAdventure

La flexibilidad de los juegos de tipo aventura gráfica como medio de transmisión de contenido educativo reside en los distintos tipos de escenas y estados que lo conforman. A continuación describimos los estados típicos de las aventuras gráficas soportados por la plataforma eAdventure:

- **Escenas interactivas:** Esta es una de las características principales de una aventura gráfica, y por lo tanto, de eAdventure. En este tipo de escenas el jugador puede, interaccionar con su entorno. De ésta forma tiene la posibilidad de recoger objetos y utilizarlos, así como dialogar con personajes o salir de dichas escenas interactivas para pasar a otras distintas.

- ***Escenas de corte:*** potencian las capacidades narrativas del motor mostrando información por la pantalla de forma no interactiva. Existen dos tipos de escenas de corte en eAdventure:
 - ***Escenas de diapositivas:*** consisten en una serie de imágenes estáticas que se van sucediendo.
 - ***Escenas de vídeo:*** consisten en escenas de reproducción de contenido audiovisual.
- ***Soporte para libros:*** las aventuras creadas para ser ejecutadas por eAdventure pueden mostrar contenido en formato texto en determinadas ocasiones. Para ello existe el objeto libro que permite mostrar por pantalla texto escrito representado en forma de un libro. De esta manera el jugador tiene la posibilidad de acceder al contenido textual, navegando por las páginas del libro como si el protagonista del juego lo estuviera leyendo.

Herramientas proporcionadas por la plataforma

A continuación se describen las herramientas proporcionadas por la plataforma de eAdventure.

Editor de juegos eAdventure

La herramienta de creación y edición de juegos proporcionada por la plataforma eAdventure facilita el proceso de creación y edición de los juegos educativos. La aplicación permite crear y configurar los distintos elementos que componen la aventura gráfica, como pueden ser escenarios, objetos, personajes o conversaciones de manera intuitiva a través de una interfaz *point-and-click*. Una vez finalizado la creación de un juego con la herramienta de autoría, puede ser exportado al formato de fichero de juegos de la plataforma caracterizado por la extensión .EAD (Véase Formato de los juegos EAD)

Si el lector desea entrar más en detalle sobre las características soportadas por la herramienta de autoría de eAdventure puede consultar la siguiente referencia:

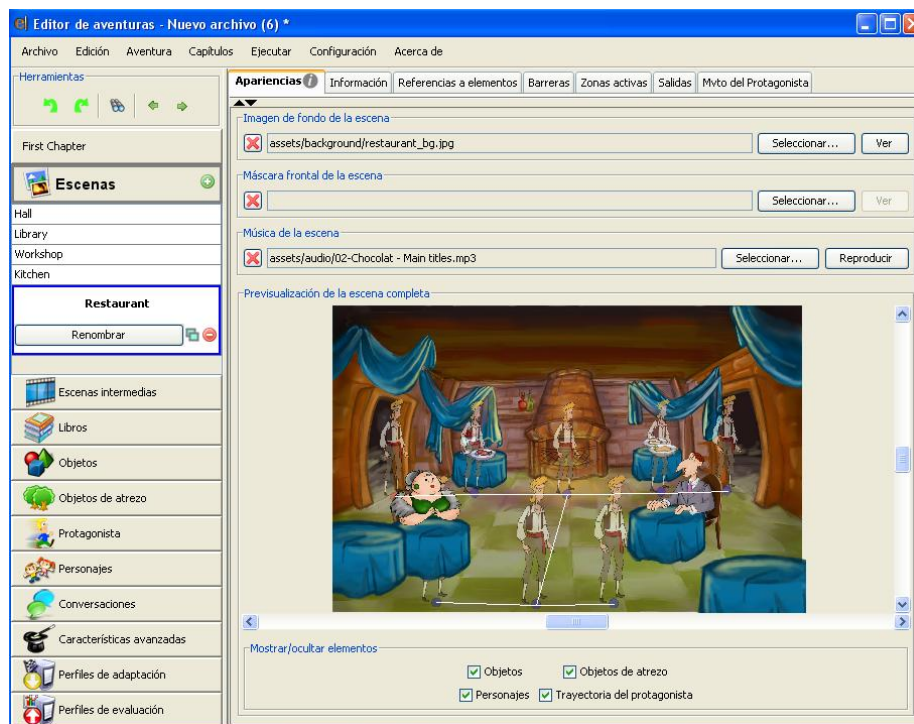


Ilustración 45 - Editor de la plataforma eAdventure

Motor de juegos eAdventure

El motor de juegos de la plataforma es el componente encargado de ejecutar los juegos realizados previamente con la herramienta de creación en entornos de escritorio y en la Web a través de *Java Applets*. El motor se encarga de cargar los juegos a través de los ficheros EAD e inicia la ejecución de cada una de las escenas definidas en el juego controlando la lógica y los distintos estados por los que transcurre el guión del juego.

En el caso de que el lector desee entrar más en detalle sobre las características del motor de escritorio, puede consultarlas en la siguiente referencia:



Ilustración 46 - Motor de juegos de eAdventure

Formato de los juegos EAD

Los ficheros EAD generados a través de la herramienta de autoría concentran toda la información para describir el comportamiento del juego. Se trata de archivos comprimidos en formato ZIP que empaquetan dicha información dividida en los elementos que se describen a continuación:

- Recursos artísticos: imágenes, sonidos y videos para mostrar por pantalla la representación del estado del juego.
- Documentos XML: Estos archivos contienen información sobre la ubicación de los recursos artísticos y los elementos narrativos de la aventura. Existen varios documentos XML cuyas sintaxis está establecida por sus correspondientes ficheros DTD. Estos archivos son:
 - Archivos con el guión del juego : El flujo narrativo de los juegos se estructura en capítulos, que son secciones del juego independientes entre sí. De esta forma son más sencillos de producir y mantener, además

reducen la carga en memoria de todo el juego. La narración de cada capítulo está descrita en un archivo XML diferente.

- **Descriptor de la aventura :** contiene información de carácter general sobre la aventura como el título, la descripción, los ajustes de la interfaz de usuario y los datos de la configuración de cada capítulo.
- **Perfiles de evaluación automática :** Cada capítulo puede tener asociado un perfil de evaluación que permite la evaluación automática de los estudiantes. Para llevar a cabo la evaluación, el profesor declarara variables que indicarán el resultado de la evaluación en un documento XML.

Tecnologías utilizadas por la plataforma

A continuación se describen las tecnologías empleadas en el desarrollo de las distintas herramientas de la plataforma.

- ***Lenguaje de programación utilizado:*** Tanto la herramienta de creación de videojuegos como el motor de eAdventure están desarrollados usando el lenguaje de programación Java en su versión 1.6.0, asegurando así su compatibilidad con los distintos sistemas operativos de escritorio.
- ***Procesamiento de información sobre los juegos:*** Como se explicó anteriormente, los juegos desarrollados por la plataforma eAdventure contienen información clave para su ejecución en archivos escritos en lenguaje de marcado XML. El motor eAdventure hace uso de SAX (*Simple API for XML*), una API dedicada al tratamiento de lenguaje XML en Java para desempeñar la tarea. Ésta herramienta facilita la verificación del contenido a través de ficheros de definición de tipo de documento DTD (Definición de tipo de documento) que contienen las descripciones de la estructura y sintaxis de los ficheros XML de las aventuras gráficas.

5.2. Análisis de Android

A continuación se realiza un estudio detallado de la plataforma móvil Android.

Introducción

Android es una pila de software para dispositivos móviles que incluye como base un sistema operativo basado en el núcleo de Linux, una capa intermedia de software (*middleware*) y aplicaciones básicas para dispositivos móviles.

La imagen inferior muestra los componentes más significativos del sistema operativo de Android dividido en cinco capas. Cada sección se explica brevemente a continuación

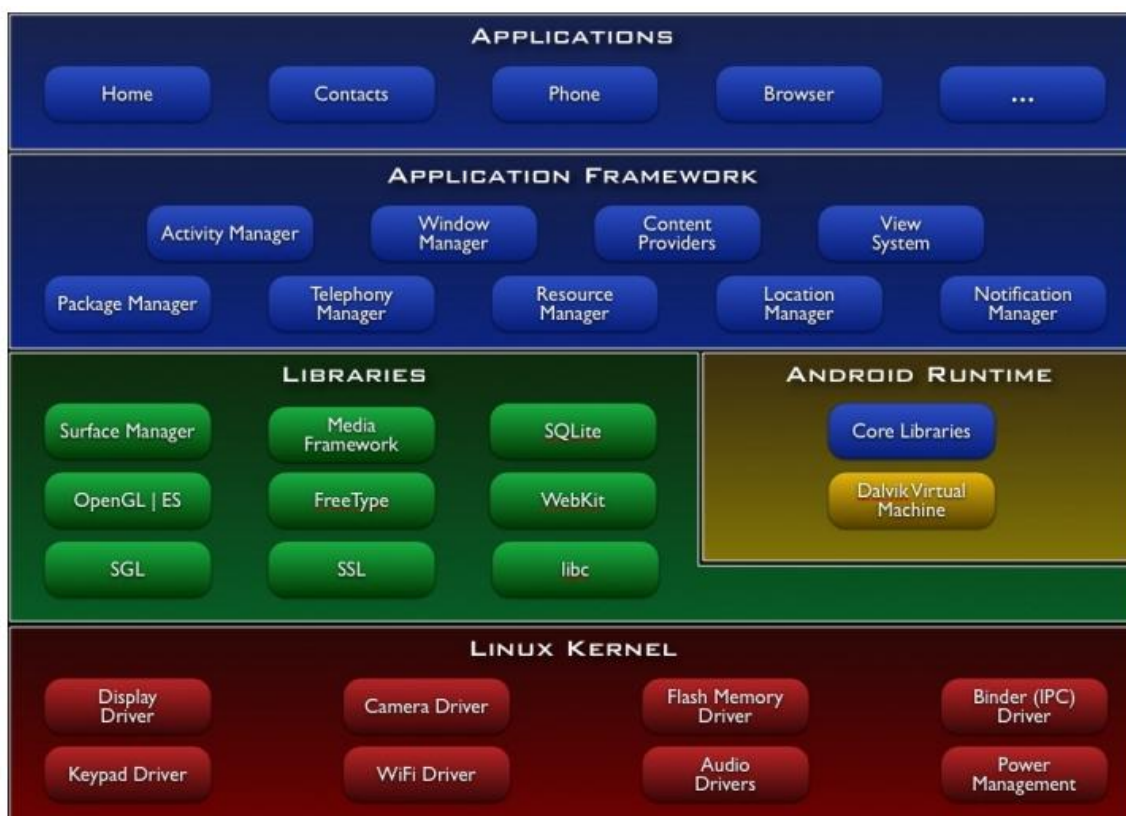


Ilustración 47 –Componentes principales de la plataforma Android

Kernel de Linux

Android se basa en la versión 2.6 del núcleo de Linux para gestionar varios servicios del sistema, como la seguridad, gestión de memoria, gestión de procesos, la capa de red o los controladores hardware. El núcleo también actúa como capa de abstracción entre el hardware y el resto de la pila de software.

Android runtime

Core Libraries: Android incluye un conjunto de bibliotecas que facilitan la mayoría de la funcionalidad proporcionada por las bibliotecas estándar de Java.

Dalvik Virtual Machine: Cada aplicación Android se ejecuta en su propio proceso y sobre una instancia nueva de la máquina virtual *Dalvik*.

La máquina virtual ejecuta clases compiladas en Java transformadas al formato *.dex*. Además ha sido diseñada para que un dispositivo pueda ejecutar varias instancias de manera eficiente. Se basa en el soporte del núcleo de Linux para la funcionalidad de multi-hilo y gestión de memoria a bajo nivel.

Bibliotecas

Android incluye un conjunto de bibliotecas C y C++ utilizadas por los componentes del sistema. Estas bibliotecas están disponibles para los desarrolladores a través del *Framework* de aplicaciones de Android.

“Framework” de aplicaciones Android

Siendo Android una plataforma de desarrollo libre ofrece a los desarrolladores la capacidad de crear aplicaciones que utilicen el hardware de los dispositivos sin limitaciones. Los desarrolladores tienen acceso a las mismas API con las que se han implementado las aplicaciones básicas de la pila software. La arquitectura del sistema está diseñada para simplificar la reutilización de componentes pudiendo acceder desde dentro de una aplicación a partes de otra aplicación que lo permita.

Aplicaciones

Por último, Android ha desarrollado algunas aplicaciones básicas y comunes en dispositivos móviles, tales como un cliente de correo electrónico, aplicaciones para enviar SMS, navegador Web o una aplicación para gestionar contactos.

Versiones de la plataforma Android

Esta sección contiene información relativa a la cantidad de dispositivos que usan las distintas versiones de la plataforma Android. Esto permite entender el escenario de la distribución de dispositivos y determinar en qué versión priorizar el desarrollo del proyecto. La información mostrada a continuación, tanto en la tabla como en el gráfico, se basa en el acceso al mercado de aplicaciones de Android recogido durante el periodo comprendido entre los días 1 y 16 de Junio del 2010.

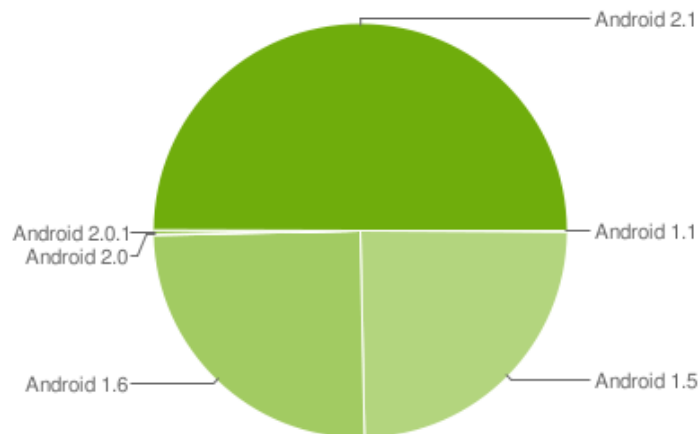


Figure 1 Porcentaje de uso de las distintas versiones de Android.

| Versión de Android | Porcentaje de dispositivos |
|--------------------|----------------------------|
| Android 1.1 | 0.1% |
| Android 1.5 | 24.6% |
| Android 1.6 | 25.0% |
| Android 2.0 | 0.1% |

| | |
|---------------|-------|
| Android 2.0.1 | 0.3% |
| Android 2.1 | 50.0% |

Dispositivos móviles Android

La plataforma Android dispone de un amplio rango de dispositivos compatibles en el mercado. Esto supone una ventaja para el despliegue de cualquier aplicación, sin embargo debido a la variedad, existen dispositivos con características diferentes que se deben tener en cuenta. Para ello se muestra a continuación una lista de algunos de los dispositivos más extendidos actualmente.



| Nombre | HTC Legend | HTC Tatto | HTC Hero | Motorola Milestone | HTC Nexus One | HTC Evo |
|----------------------------------|---------------------|-----------------|-----------------|----------------------|--------------------------|--------------------------|
| Procesador | Qualcomm MSM 600Mhz | Qualcomm 528Mhz | Qualcomm 528Mhz | 550Mhz ARM Cortex A8 | Qualcomm Snapdragon 1Ghz | Qualcomm Snapdragon 1Ghz |
| Memoria RAM | 512Mb | 512Mb | 512Mb | 512Mb | 512Mb | 1024Mb |
| Resolución de pantalla (píxeles) | 320x480 | 240x320 | 320x480 | 480x854 WVGA | 480x800 | 480x800 |
| Tamaño de pantalla (pulgadas) | 3,2 | 2,8 | 3,2 | 3,7 | 3,7 | 4,3 |
| Cámara | Si | Si | Si | Si | Si | si |
| GPS | Si | Si | Si | Si | Si | si |
| Pantalla táctil | Si | Si | Si | Si | Si | Si |
| Teclado | No | No | No | Si | No | no |

Como se puede apreciar, existen ciertas características estandarizadas en éste tipo de móviles como la pantalla táctil, la disponibilidad de sensores GPS y la cámara de fotos. Sin embargo características como el teclado serán descartadas en el desarrollo del proyecto puesto que limitan la compatibilidad con un menor número de dispositivos.

Conceptos básicos de Android

Android tiene cuatro tipos de componentes básicos que pueden estar presentes en las distintas aplicaciones

Activity

Los *Activities* son los componentes encargados de presentar la interfaz gráfica al usuario. Por ejemplo, una aplicación de mensajes de texto puede tener un *Activity* que muestra una lista de contactos de manera que cuando se selecciona uno de ellos se inicia otro *Activity* que proporciona un area de texto donde poder escribir el mensaje y enviarlo. A pesar de que existe comunicación entre los *Activities* cada uno de ellos es independiente de los demás.

Cada *Activity* dispone de un marco en el que pintar y que ocupa por defecto el tamaño de la pantalla. La disposición de los elementos gráficos se configura a través de un sistema de jerarquias de vistas como se explica más adelante en el documento.

Services

Los *Services* son componentes básicos que no disponen de interfaz gráfica y están diseñados para ser ejecutados en segundo plano. Por ejemplo, un componente *Service* puede emplearse para reproducir música de fondo, obtener datos de la red o hacer algún cálculo y proporcionar el resultado a un componente visual como puede ser un componente *Activity*. El tiempo de vida del componente *Service* depende del proceso que lo creó cerrandose cuando el proceso padre acaba o lo cierra.

Broadcast receivers

Los componentes *Broadcast Recievers* reciben y responden a mensajes globales (*broadcast*) del sistema emitidos, por ejemplo, cuando el nivel de batería es bajo o cuando se desmonta la tarjeta de memoria externa. Este tipo de mensajes pueden producir notificaciones informativas al usuario que se añaden a la barra de notificaciones del sistema.

Content provider

El componente *Content Provider* tiene como objetivo hacer accesible los datos de las aplicaciones a otras aplicaciones pertenecientes a la plataforma. Los datos pueden ser almacenados en el sistema de archivos, en una base de datos *SQLite* o en cualquier otro medio disponible.

Intents

Los *Intents* se utilizan en Android para realizar la comunicacion entre los distintos componentes básicos. Un *Intent* se puede utilizar tanto para iniciar un *Activity* como para comunicarse con *Services* que se ejecutan en segundo plano, o para enviar información a los componentes *Broadcast receivers*.

El componente Activity

En esta sección se entra más en detalle en las características principales del componente básico más utilizado, el componente *Activity*. Normalmente en una aplicación desarrollada para Android existe más de un *Activity* por cada aplicación, pero solo uno puede estar ejecutandose en primer plano en cada momento. El contenido visual de cada componente *Activity* esta proporcionado por una jerarquía de vistas (*View Hierarchy*) como el ejemplo que se muestra en el siguiente diagrama.

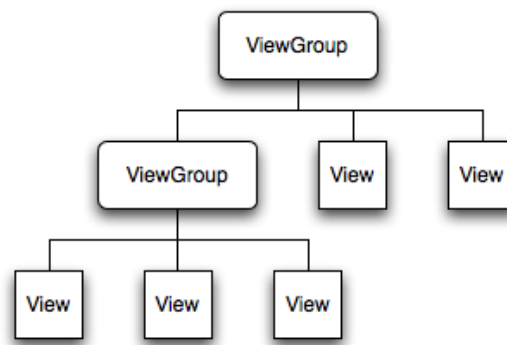


Ilustración 48 - Ejemplo de una jerarquía de vistas en Android

Los árboles de jerarquía de vistas están compuestos por distintos tipos de vistas que comprenden desde elementos simples como etiquetas de texto o botones hasta grupos compuestos de vistas como listas u otras vistas personalizadas.

Las vistas “hijas” solicitan un tamaño y ubicación dentro de la ventana de pintado, pero la vista padre se encarga de definir el tamaño final de cada hijo. Android analiza los elementos de la vista en orden (desde la parte superior del árbol de jerarquía). Si existen vistas que se superponen, el último elemento que se genera será el que se muestre por encima de los demás definidos previamente.

La manera más común en Android para definir la disposición de los elementos es a través de un archivo XML de jerarquía de vistas. Cada elemento del fichero XML es o bien una vista o bien un grupo de vistas, donde las vistas son las hojas en el árbol y los grupos de vistas son ramas en el árbol (véase la ilustración 49). A continuación se muestra un ejemplo del fichero XML de definición del *layout*:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
    </LinearLayout>

```

Ilustración 49 - Ejemplo de fichero de disposición de vistas

Ciclo de vida del componente Activity

El componente *Activity* tiene el ciclo de vida definido en la siguiente ilustración.

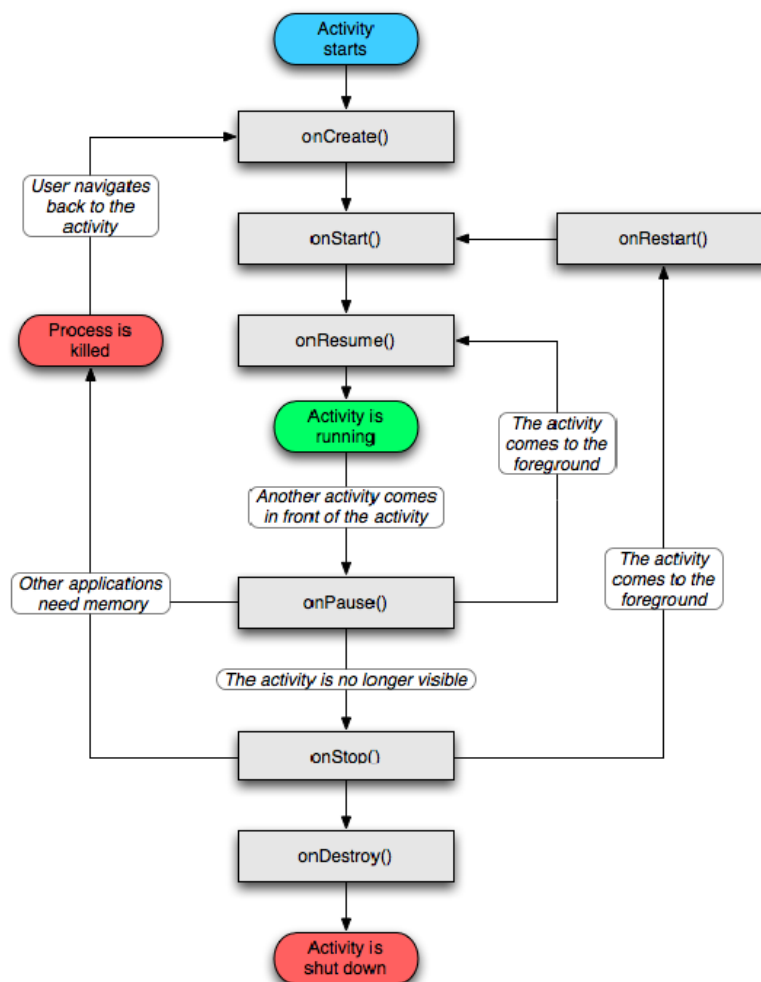


Ilustración 50 - Ciclo de vida del componente Activity en Android

Los tres estados fundamentales del ciclo de vida del componente son:

- *Activo o en ejecución:* cuando se muestra en primer plano, es decir, cuando ocupa la primera posición en la pila de *Activities* de la “tarea” actual. Éste es el *Activity* que recibe las acciones del usuario.
- *Pausa:* cuando el *Activity* pierde el foco pero sigue siendo visible para el usuario. Esto sucede cuando otro *Activity* se encuentra superpuesto. Cuando está en estado pausado el *Activity* continúa “vivo” (conserva la información y sigue unido al gestor de ventanas), pero puede ser eliminado por el sistema cuando la memoria disponible sea escasa.
- *Detenido:* el componente pasa a este estado cuando queda oculto por otro *Activity*. Conserva el estado e información asociada, sin embargo, ya no es visible para el usuario por lo que su ventana se oculta. El sistema puede eliminar el *Activity* cuando otros procesos necesiten recursos de memoria.

El ciclo de vida completo de un *Activity* transcurre desde su creación a través del método *OnCreate()*, donde se realiza toda su configuración inicial, hasta la llamada final a *OnDestroy()*, donde se liberan todos los recursos que tiene asignados. Un *Activity* es visible al usuario entre las llamadas a *OnStart()* y *onStop()*. Los *Activity* pueden pasar con frecuencia al estado *onPause()*, por ejemplo, cuando el dispositivo móvil pasa a estado de suspensión o cuando otro *Activity* pasa a primer plano.

Por último siempre que un *Activity* haga una llamada a *onStop()* se ocultará, y muy probablemente será eliminado por el sistema.

Gráficos en Android

Los gráficos en Android se gestionan a través de la librería gráfica 2D, en el caso de gráficos en dos dimensiones, y OpenGL 1.0 de alto rendimiento para gráficos en tres dimensiones. Al desarrollar una aplicación en Android, es importante tener en cuenta el tipo de aplicación a desarrollar. En el caso de ser una aplicación que hace uso de gráficos en dos dimensiones, la plataforma plantea las siguientes soluciones:

- Dibujar gráficos o animaciones dentro de un objeto *View*, de manera que el dibujo o animación es gestionado por el sistema de jerarquía de vistas, y por lo tanto el refresco de pintado depende de dicho sistema.
- Dibujar los gráficos directamente en sobre un objeto de tipo “lienzo”. De esta manera, el desarrollador es el encargado de llamar al método de pintar *Draw()* de la clase apropiada. Esta opción es más adecuada en caso que la aplicación realice un uso intensivo del pintado. *SurfaceView* es la clase de tipo lienzo. Es una subclase de la clase *View* que ofrece una superficie de dibujo específica dentro de la jerarquía de vistas. El objetivo es ofrecer dicha superficie de dibujo a un subproceso de la aplicación de manera que el pintado de gráficos no es gestionado por el sistema de jerarquías de vistas. En su lugar, el subproceso encargado del pintado pide el lienzo al objeto *SurfaceView* para dibujar con la frecuencia deseada.

Formatos multimedia

En Android los ficheros multimedia pueden estar alojados directamente dentro de los recursos de la aplicación, como archivos independientes en el sistema de archivos del dispositivo o acceder a ellos a través de un flujo de datos en una conexión de red (*streaming*).

A continuación se muestra la relación de formatos de ficheros multimedia soportados por la plataforma.

| Type | Format | Encoder | Decoder | Details | File Type(s) Supported |
|-------|--------------------------|---------|---------|--|---|
| Audio | AAC LC/LTP | | X | Mono/Stereo content in any combination of standard bit rates up to 160 kbps and sampling rates from 8 to 48kHz | 3GPP (.3gp) and MPEG-4 (.mp4, .m4a). No support for raw AAC (.aac) |
| | HE-AACv1 (AAC+) | | X | | |
| | HE-AACv2 (enhanced AAC+) | | X | | |
| | AMR-NB | X | X | 4.75 to 12.2 kbps sampled @ 8kHz | 3GPP (.3gp) |
| | AMR-WB | | X | 9 rates from 6.60 kbit/s to 23.85 kbit/s sampled @ 16kHz | 3GPP (.3gp) |
| | MP3 | | X | Mono/Stereo 8-320kbps constant (CBR) or variable bit-rate (VBR) | MP3 (.mp3) |
| | MIDI | | X | MIDI Type 0 and 1. DLS Version 1 and 2. XMF and Mobile XMF. Support for ringtone formats RTTTL/RTX, OTA, and iMelody | Type 0 and 1 (.mid, .xmf, .mxmf). Also RTTTL/RTX (.rtttl, .rtx), OTA (.ota), and iMelody (.imy) |
| | Ogg Vorbis | | X | | Ogg (.ogg) |
| | PCMWAVE | | X | 8- and 16-bit linear PCM (rates up to limit of hardware) | WAVE (.wav) |
| Image | JPEG | X | X | Base+progressive | JPEG (.jpg) |
| | GIF | | X | | GIF (.gif) |
| | PNG | | X | | PNG (.png) |
| | BMP | | X | | BMP (.bmp) |
| Video | H.263 | X | X | | 3GPP (.3gp) and MPEG-4 (.mp4) |
| | H.264 AVC | | X | | 3GPP (.3gp) and MPEG-4 (.mp4) |
| | MPEG-4 SP | | X | | 3GPP (.3gp) |

Ilustración 51- Formatos de ficheros multimedia soportados por la plataforma Android

Soporte multi-pantalla

Android está diseñado para ser ejecutado en una gran variedad de dispositivos móviles. Estos dispositivos tienen pantallas de distintos tamaños y resoluciones. La plataforma tiene mecanismos que ayudan a tratar la heterogeneidad de las pantallas. De esta manera los desarrolladores, con una sola aplicación *APK*, dan soporte a toda la diversidad de pantallas soportadas por Android.

En cuanto al soporte de múltiples pantallas en Android cabe destacar que hay un antes y un después a partir de la versión 1.6 del sistema operativo. En versiones anteriores a la versión 1.5 (inclusive) solo se da soporte a un tipo de pantalla con un tamaño y resolución único. Es a partir de la actualización 1.6 cuando se da soporte a múltiples pantallas con resoluciones distintas.

En las siguientes secciones se repasan los distintos aspectos a tener en cuenta a la hora de crear una aplicación en Android con soporte para distintas configuraciones de

pantalla, empezando por una breve introducción de los términos y conceptos básicos de las pantallas de dispositivos móviles.

Tamaño de pantalla: tamaño físico de la pantalla, considerando la medida real de la diagonal de la pantalla. Android para simplificar divide el tamaño de las pantallas en tres grupos, grande, normal y pequeño. Los recursos gráficos y su disposición se ajustan al tamaño de pantalla de manera transparente al desarrollador siempre que se use su sistema de *jerarquía de vistas*.

Relación de aspecto: La relación proporcional entre el ancho físico y el alto físico de la pantalla. Las aplicaciones pueden contener recursos específicos para las distintas proporciones de pantalla (*Aspect ratio*) usando calificadores de recursos: *long* y *not long*.

Resolución de pantalla: El número de píxeles físicos de la pantalla. Se mide en número de píxeles de ancho y número de píxeles alto.

Densidad de píxeles (dpi): La densidad de píxeles hace referencia a la idea de que en dispositivos diferentes el tamaño físico de los píxeles puede diferir. De esta manera al renderizar los elementos gráficos como, por ejemplo, un botón en función de píxeles físicos se visualiza más largo en la pantalla con menor densidad de píxeles que en la de mayor densidad. Debido a la cantidad de configuraciones distintas posibles, Android generaliza las distintas densidades de pantallas en alta, media o baja. Las aplicaciones pueden contener recursos distintos para cada densidad de píxeles, con el fin de que se visualicen bien en todas las pantallas.

Densidad independiente de píxeles (dip): El mecanismo que utiliza Android para abstraer al desarrollador de la densidad de píxeles de la pantalla, es la densidad de píxeles independiente del dispositivo. Cada píxel medido en *dips* es un píxel virtual que representa un píxel físico real de la pantalla definida como base en Android : 420x360 píxeles y densidad píxeles 160dips.

En ejecución la plataforma puede encargarse transparentemente de escalar el tamaño y posición de los recursos definidos con esta métrica. Solo se encarga de escalar los recursos definidos dentro de la aplicación, en cambio si los recursos los gestiona el desarrollador en tiempo de ejecución, el desarrollador tendrá que tratarlo personalmente con ayuda de otros componentes del *framework*.

Para calcular el número de píxeles reales a partir de los *dips* en una configuración de pantalla dada es sencillo y basta con aplicar la siguiente ecuación

$$\text{Píxeles reales} = \text{DIPS} * (\text{Densidad} / 160)$$

Por ejemplo, en una pantalla de 240 píxeles físicos reales, un píxel independiente equivaldría a 1,5 píxeles reales.

Rango de pantallas soportado

Como se comentó anteriormente existe un antes y un después en el soporte de distintas configuraciones de pantalla en Android. En versiones de la API anteriores a la 1.5 (inclusive) se daba soporte tan solo a una configuración de pantalla HVGA (320x480 píxeles) de 3,2 pulgadas y densidad de píxeles 160 dpi.

A partir de la versión 1.6 de la API de Android se añade soporte multi-pantalla. Para simplificar la tarea al desarrollador se crearon grupos de pantallas dependiendo de sus características. Se divide en tres grupos generales de tamaño y otros tres grupos generales de densidades. La siguiente tabla muestra las configuraciones de pantalla soportadas actualmente:

| | Low density (120), <i>ldpi</i> | Medium density (160), <i>mdpi</i> | High density (240), <i>hdpi</i> |
|----------------------|---|---|---|
| <i>Small screen</i> | <ul style="list-style-type: none"> QVGA (240x320), 2.6"-3.0" diagonal | | |
| <i>Normal screen</i> | <ul style="list-style-type: none"> WQVGA (240x400), 3.2"-3.5" diagonal FWQVGA (240x432), 3.5"-3.8" diagonal | <ul style="list-style-type: none"> HVGA (320x480), 3.0"-3.5" diagonal | <ul style="list-style-type: none"> WVGA (480x800), 3.3"-4.0" diagonal FWVGA (480x854), 3.5"-4.0" diagonal |
| <i>Large screen</i> | | <ul style="list-style-type: none"> WVGA (480x800), 4.8"-5.5" diagonal FWVGA (480x854), 5.0"-5.8" diagonal | |

Ilustración 52 - Características de las distintas pantallas soportadas por la plataforma Android

De esta manera la aplicación puede contener recursos específicos para cada tipo de pantalla y la plataforma se encarga de cargar los recursos de forma transparente al desarrollador, adaptándolos a las características físicas del dispositivo. Estos recursos se definen a través de marcadores XML.

6. Análisis del sistema

Tras haber analizado las plataformas en las que se basa el desarrollo del proyecto y haber estudiado distintas fórmulas de adaptación de juegos educativos en entornos móviles, surge la necesidad de definir las incorporaciones a realizar en la plataforma eAdventure para dar soporte a juegos educativos para Android.

En el siguiente apartado se definen las funcionalidades a integrar en todas las fases del proceso de creación, distribución y ejecución de juegos en la plataforma eAdventure. A continuación se analizan posibles riesgos y se realizan prototipos que demuestren la viabilidad de las incorporaciones a la plataforma.

Ampliación de la plataforma eAdventure

Motor de juegos para Android

En primer lugar se debe crear un nuevo motor de juegos para Android.

A continuación se define la funcionalidad que debe soportar el nuevo motor de juegos eAdventure para Android. Se ha realizado una captura de requisitos del motor de juegos de escritorio de cara a dar soporte a las mismas funcionalidades. Por otro lado se han incluido los requisitos funcionales relacionados con el soporte de ejecución de los nuevos tipos de juegos geolocalizados y basados en códigos QR.

Requisitos funcionales del cargador de juegos

Carga de juegos

| | |
|---------|---|
| RFCJ-00 | Debe permitir la carga de juegos eAdventure clásicos, geolocalizados y basado en códigos QR en el núcleo del motor permitiendo iniciar una partida jugable. |
|---------|---|

Carga de partidas

| | |
|---------|---|
| RFCP-00 | Debe permitir la carga de partidas previamente guardadas.[RFCP-01] |
| RFCP-01 | La información asociada a cada partida guardada debe contener el nombre del juego, el nombre de salvado y una captura del juego en ejecución en el momento del guardado |

Instalación de juegos

| | |
|---------|---|
| RFIJ-00 | Debe permitir la instalación de juegos previamente creados y exportados para Android con la herramienta de autoría de eAdventure [RFAUT-00] |
| RFIJ-01 | La instalación de un juego debe poder realizarse desde un fichero ubicado en memoria externa o tras haberlo descargado haciendo uso del repositorio de juegos [RFRP-02] |
| RFIJ-02 | Debe permitir la desinstalación de juegos previamente instalados |

Repositorio de juegos

| | |
|---------|--|
| RFRP-00 | Debe permitir la consulta de los juegos incluidos en el repositorio de eAdventure ubicado en http://code.google.com/p/eadventure-android |
|---------|--|

| | |
|---------|--|
| RFRP-01 | Debe permitir consultar la información asociada a cada juego incluido en el repositorio [RFRP-00]. Debe incluir el nombre del juego, una imagen que represente el juego y una breve descripción de la temática del juego |
| RFRP-02 | Debe permitir la descarga e instalación de los juegos incluidos en el repositorio, siempre que exista conectividad a Internet. |

Preferencias

| | |
|--------|--|
| RFP-00 | Debe permitir activar o desactivar el audio durante la ejecución de los juegos [RFOP-02] |
| RFP-01 | Debe permitir activar o desactivar la vibración asociada a la interacción durante el juego |
| RFP-03 | Debe permitir la instalación de juegos previamente guardados en la tarjeta de memoria del dispositivo móvil |
| RFP-04 | Debe permitir acceder a la página Web oficial del proyecto eAdventure ubicada en http://e-adventure.e-ucm.es/ |

Funcionalidades del núcleo del motor de juegos

A continuación se muestran las principales funcionalidades a cubrir de la lógica del motor de juegos de eAdventure.

Opciones en ejecución

| | |
|---------|---|
| RFOP-00 | Se debe permitir salvar la partida durante la ejecución del juego, en cuyo caso se indica el nombre de la partida [RFOP-04], más adelante esa partida debe aparecer en la carga de partidas [RFCP-00] |
| RFOP-01 | Debe permitir activar o desactivar el audio durante la ejecución del juego |
| RFOP-02 | Debe permitir salir de la partida en curso, en cuyo caso debe controlar el |

| | |
|---------|---|
| | guardad de la partida [RFCJ-00] |
| RFOP-03 | Debe permitir redimensionar el tamaño de la escena de juego adaptándose por completo al tamaño de la pantalla [RFV-00] y [RFV-01] |

Visualización de los juegos

| | |
|--------|--|
| RFV-00 | La escena se muestra escalada al alto de la pantalla del dispositivo (en orientación apaisada) |
| RFV-01 | Puede ajustarse la imagen del juego al tamaño de la pantalla (sin mantener proporción) |

Juegos geolocalizados

| | |
|----------|--|
| RFJG--00 | En caso de cargar un juego de tipo geolocalizado la señal GPS debe estar activada. Si no, debe gestionar la activación de ésta para continuar con la ejecución del juego [RFAV-00] |
|----------|--|

Juegos con códigos QR

| | |
|----------|--|
| RFJG--01 | Permiten lanzar uno o mas efectos[RFEF-00] si se han cumplido las condiciones[RFCF-00] |
|----------|--|

Escenas

| | |
|--------|---|
| RFE-00 | Pintar los recursos de fondo de escena asociados a la escena cargada. Los demás elementos pintados deben estar situados sobre él.[RFE-01] |
| RFE-01 | Pintar objetos de primer plano encima del resto del fondo [RFE-00] |

| | |
|--------|--|
| RFE-02 | Pintar máscara frontal de la escena encima de todos los elementos pintados. [RFE-01] |
| RFE-03 | Mostrar los elementos asociados a la escena si se cumplen las condiciones asociadas a cada elemento [RFCF-00] |
| RFE-04 | Mostrar los personajes en una su posición inicial definida en la escena |
| RFE-05 | Evitar el movimiento de los personajes a través de las barreras definidas en las escenas. |
| RFE-06 | Los personajes tiene trayectorias por las que pueden desplazarse a través de la escena |
| RFE-07 | Las escenas pueden tener zonas activas o interactivas que pueden tener acciones asociadas [RFA-02] |
| RFE-08 | Las escenas pueden tener salidas que producen un cambio a la escena definida. |
| RFE-09 | El protagonista puede interactuar con los objetos de la escena. Los objetos tienen acciones asociadas [RFO-03] |
| RFE-11 | El protagonista puede interactuar con los personajes de la escena [RFA-01] |

Objetos

| | |
|--------|--|
| RFO-00 | Los objetos son elementos que pueden estar asociados a las escenas |
| RFO-01 | Deben tener una posición relativa dentro de la escena asociada |
| RFO-02 | Debe tener una serie de acciones asociadas [RFA-00] |

Inventario

| | |
|--------|--|
| RFE-10 | El inventario contiene los objetos [RFO] que el usuario ha cogido previamente. |
|--------|--|

Acciones

| | |
|--------|---|
| RFA-00 | Los objetos se pueden coger, examinar y usar |
| RFA-01 | Los personajes se pueden examinar, hablar y usar con |
| RFA-02 | Las zonas activas se pueden coger, examinar, usar y usar con |
| RFA-03 | Se pueden tener acciones personalizadas con los objetos, zonas Activas y personajes |
| RFA-04 | Las acciones pueden lanzar efectos [RFEF] |
| RFA-05 | Puede tener asociados 0..N efectos [RFEF] |

Libros

| | |
|--------|--|
| RFL-00 | Los libros pueden ser leídos por el protagonista del juego |
| RFL-01 | Muestran información textual |

Objetos de atrezzo

| | |
|---------|--|
| RFOA-00 | Los objetos de atrezzo no permiten ningún tipo de interacción por parte del protagonista del juego |
| RFOA-01 | Se muestran como objetos en escenas en una posición definida. |

Personajes

| | |
|--------|--|
| RFP-00 | Los personajes pueden tener 0..N apariencias distintas |
| RFP-01 | Las apariencias cambian según la condiciones establecidas [RFCF] |
| RFP-02 | Las apariencias se dividen en grupos (hablando, caminando, parado, usando) |
| RFP-03 | Cada grupo de apariencias se aplica al lanzar la acción asociada |
| RFP-05 | El dialogo entre personajes puede tener un bocadillo asociado [RFC-02] |
| RFP-06 | En dialogo el personaje tiene un color de letra y un color de fondo |

| | |
|--------|--|
| | definido |
| RFP-07 | El protagonista se debe aproximar a los elementos cuando interactúa con ellos [RFA-01] |

Conversaciones

| | |
|--------|---|
| RFC-02 | Los diálogos contienen líneas de texto que pueden tener bocadillos asociados [RFP-05] |
| RFC-03 | Cada línea de texto debe pertenecer al personaje que se encuentra hablando en el momento de la ejecución |
| RFC-04 | Cada línea se debe mostrar durante un tiempo proporcional a su longitud |
| RFC-06 | Se debe permitir la elección de las opciones en la conversación por parte del jugador |
| RFC-07 | Cuando el jugador selecciona una opción, el flujo de la conversación continúa por el nodo referenciado por dicha opción |
| RFC-08 | Las conversaciones pueden tener audio asociado dependiendo de si esta activado o no en la opciones [RFOP-02] |
| RFC-09 | Las líneas de la conversación pueden tener condiciones asociadas para ser mostradas [RFCF] |
| RFC-11 | Los nodos de la conversación pueden tener efectos asociados [RFEF] |

Condiciones, flags y variables

| | |
|---------|--|
| RFCF-00 | Los condiciones deben ser sentencias lógicas que se evalúan como cierto o falso |
| RFCF-01 | Las condiciones pueden contenerlos operadores lógicos AND y OR |
| RFCF-02 | Las condiciones verifican fórmulas compuestas por valores de <i>flags</i> , variables y estados globales |
| RFCF-03 | Los <i>flags</i> tienen dos estados, activados o desactivados |
| RFCF-04 | Las variables tienen un nombre y se les asocia un valor numérico |

| | |
|---------|--|
| RFCF-05 | Se puede aplicar operadores de comparación sobre variables (<,>,<=,>=,=) |
| RFCF-06 | Los estados Globales son equivalentes a <i>flags</i> asociados a un estado general del juego |

Efectos

| Identificador | Descripción |
|---------------|---|
| RFEF-00 | Los efectos son instrucciones que actúan durante la ejecución del juego |
| RFEF-01 | Son lanzados cuando se cumplen las condiciones asociadas (si las hubiera) [RFCF-00] |
| RFEF-02 | Permiten activar y desactivar <i>flags</i> definidos en el capítulo [RFCF-03] |
| RFEF-03 | Permiten cambiar el valor de variables del capítulo [RFCF-04] |
| RFEF-04 | Permiten iniciar la reproducción de audio |
| RFEF-05 | Pueden lanzar una animación del capítulo, en una posición específica |
| RFEF-06 | Pueden lanzar “Escenas intermedias”, Diálogos de protagonista o personajes, conversaciones, libros |
| RFEF-07 | Permiten consumir objetos (eliminarlos del escena) |
| RFEF-08 | Permiten generar objetos (añadirlos al inventario) |
| RFEF-09 | Permiten mover cualquier personaje a una nueva posición en la escena |
| RFEF-10 | Permiten cancelar los efectos que lanzan acciones |
| RFEF-11 | Se puede definir la probabilidad de ser lanzado para dos efectos, y lanzar uno de ellos según el azar (ambas probabilidades sumarán 100%) |
| RFEF-12 | Lanzar espera de tiempo que bloquea el juego durante ciertos segundos |
| RFEF-13 | Una Macro es una secuencia de efectos ordenados |
| RFEF-14 | Un efecto puede lanzar una macro |

Escenas Intermedias

| | |
|---------|---|
| RFEI-00 | Tipo de escena sin interacción que muestra contenido al usuario |
|---------|---|

| | |
|---------|---|
| RFEI-01 | Pueden ser de diapositivas o de video |
| RFEI-02 | Ocupan toda la pantalla al mostrarse |
| RFEI-03 | Si son diapositivas muestran una animación en formato “Diapositivas JPG” o EAA, y pueden reproducir un sonido en formato MP3. |
| RFEI-04 | Los videos reproducen los formatos MP4. |
| RFEI-05 | Al finalizar pueden volver a la escena anterior, ir a una nueva escena, o terminar el capítulo |

Características avanzadas Temporizadores

| | |
|----------|--|
| RFCAT-00 | Los temporizadores tienen un tiempo asociado entre 1 y 356.400 segundos |
| RFCAT-01 | Se pueden mostrar por pantalla opcionalmente (nombre, “avanzando” o “retrocediendo”, y segundos) |
| RFCAT-02 | Se activan cuando se cumplen sus condiciones iniciales [RFCF-00] |
| RFCAT-03 | Si no tiene condiciones definidas se activa automáticamente |
| RFCAT-04 | Pueden tener condiciones finales que lo desactivan cuando se cumplen. |
| RFCAT-05 | Pueden lanzar un efecto [RFEF-00] cuando el tiempo expira o cuando se desactiven |
| RFCAT-06 | Pueden iniciarse una vez para si se cumplen las condiciones de entrada |
| RFCAT-07 | Pueden funcionar en bucles o detenerse al llegar a su fin |

Estado globales

| | |
|---------|---|
| RFEG-00 | Los EG tienen un identificador y una lista de condiciones |
| RFEG-01 | Un EG estará activo si se cumplen todas sus condiciones o si no tiene condiciones |

Cambios en la herramienta de autoría

Con el fin de proporcionar soporte para la creación de juegos Android en eAdventure, se debe definir los cambios a realizar en la herramienta de autoría de juegos teniendo en cuenta los siguientes objetivos:

- La creación de juegos debe permitir su despliegue en las plataformas de escritorio, Web y móviles Android.
- Los juegos anteriormente creados con la plataforma deben ser compatibles con dispositivos móviles Android (Compatibilidad hacia atrás).
- El proceso de creación de los juegos educativos en eAdventure NO debe ser modificado.

La tabla inferior muestra los requisitos funcionales a integrar en la herramienta de autoría de eAdventure

| | |
|----------|---|
| RFAUT-00 | Permitir la exportación de juegos para la plataforma Android previa comprobación de la compatibilidad de los recursos multimedia (Perfil de exportación) [RFAUT-01] |
| RFAUT-01 | Comprobación de la compatibilidad de recursos multimedia entre la plataforma eAdventure y Android. |
| RFAUT-02 | Mostrar informe de comprobación de compatibilidad de recursos multimedia teniendo en cuenta las restricciones definidas en [RFAUT-01] |

Prototipos

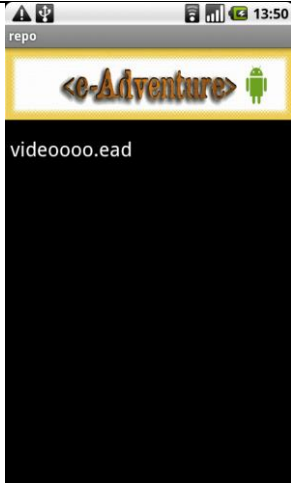
A continuación se muestran los prototipos creados durante el desarrollo del proyecto.

Prototipo de carga del modelo de juegos eAdventure

| | |
|-------------------|--|
| Titulo | Carga del modelo de juegos eAdventure |
| Alcance | ZIP , XML , Generación instancias del modelo |
| Objetivo | Demostrar la viabilidad y funcionamiento de la carga del modelo de datos de los juegos creados con la plataforma eAdventure |
| Descripción | Prototipo capaz de realizar la lectura del modelo de datos XML desde la tarjeta de memoria externa <i>SDCard</i> utilizando la API de SAX proporcionada por Android . |
| Aspectos técnicos | El prototipo debe determinar la funcionalidad la API de SAX proporcionada por la API de Android, comprobar si tiene alguna limitación, medir tiempos de carga del modelo de datos del juego, comprobar las limitaciones del acceso al fichero EAD. |
| Resultados | <p>La API es funcional pero no valida definiciones de documentos DTD con lo que los valores escritos por defecto en los ficheros DTD deben ser introducidos manualmente en caso de no encontrar el nodo correspondiente en el fichero XML. Tampoco tiene soporte para espacios de nombres. Esto no es imprescindible pero ha requerido la adaptación de varias partes del código.</p> <p>La carga es rápida y el modelo de datos se genera sin dificultad.</p> <p>El acceso al fichero ZIP se realiza sin problemas.</p> <p>En caso de extracción de tarjeta de memoria se controla el acceso y se informa de que no es accesible.</p> |

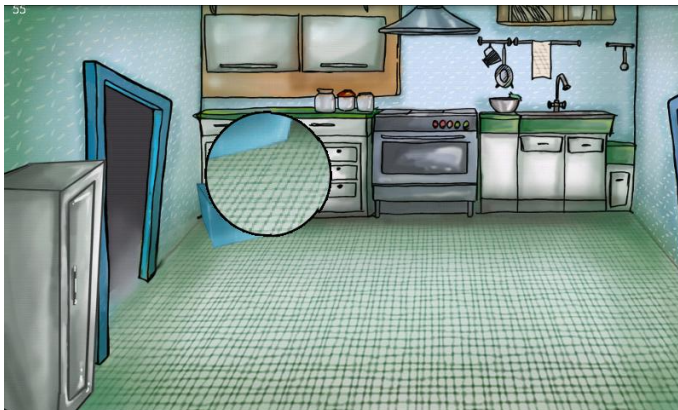
Prototipo de repositorio de juegos

| | |
|-------------------|---|
| Titulo | Repositorio de juegos |
| Alcance | Conectividad a Internet, descarga, almacenamiento, sistema repositorio. |
| Objetivo | Demostrar la viabilidad del desarrollo de un sistema de distribución de juegos a través de Internet basado en repositorio de descargas. |
| Descripción | Prototipo capaz de descargar un juego y guardarlo en memoria externa del dispositivo. Comprobamos la funcionalidad existente dentro de la API de Android para el manejo de eventos externos, como controlar errores, cortes de descarga o descargas completadas. |
| Aspectos técnicos | Se debe comprobar la viabilidad de la descarga de juegos educativos eAdventure usando los distintos tipos de conectividad disponibles por la plataforma Android. Se debe medir las velocidades de descarga a través de cada medio y controlar los errores provocados por cortes en la conexión de la red. |
| Resultados | Sí, es viable la descarga de contenido. La velocidad varía entre las distintas tecnologías de transmisión de datos. Los juegos de eAdventure ocupan una media de 30Mb, con lo que debería ser viable. En caso de que la conexión sea lenta, la descarga será detenida. El prototipo informa de un error cuando se produce una desconexión de red y borra los datos descargados hasta ese momento. |

| | |
|------------------|--|
| Captura pantalla |  |
|------------------|--|


Prototipo del método de pintado

| | |
|-------------------|--|
| Titulo | Método de pintado |
| Alcance | Diseño de la estructura de pintado, refresco de pantalla, carga de imágenes desde sistema de ficheros |
| Objetivo | Demostrar la viabilidad del sistema de pintado de Android, su tiempo de refresco dentro de un diseño específico acoplable al motor de juegos. Escalado y pintado de las imágenes cargadas |
| Descripción | Prototipo capaz de pintar un fondo de pantalla y pintar un personaje animado que va desplazándose por la pantalla. Comprobamos la funcionalidad existente dentro de la API de Android para el refresco de la pantalla, la construcción de la imagen final y la carga desde un imagen desde una carpeta comprimida |
| Aspectos técnicos | Se debe estudiar la viabilidad del uso de una superficie de pintado de Android a través de un hilo de ejecución externo al Activity que la posee. Frecuencia de actualización y eficiencia del pintado. Posibilidad de carga de imágenes desde archivos comprimidos. Rendimiento de Android en el escalado de imágenes. Necesidad de caches de imágenes. |

| | |
|----------------------|--|
| Resultados aportados | <p>Sí, es viable el diseño de la aplicación donde el pintado se realiza desde un hilo externo a su <i>Activity</i>. El refresco de la imagen es adecuado. La animación del personaje se muestra con un movimiento fluido sin mostrar retardos en ningún momento. El escalado no parece afectar el rendimiento del sistema en exceso. Además se puede cargar imágenes desde un archivo comprimido. La construcción de una caché no es necesaria puesto que la carga desde el sistema de ficheros es muy rápida.</p> |
| Captura pantalla |  A screenshot from a game showing a kitchen environment. The kitchen has green walls, a green tiled floor, and white cabinetry. A circular inset highlights a specific area of the floor. The scene includes a stove, a sink, and various kitchen items. |

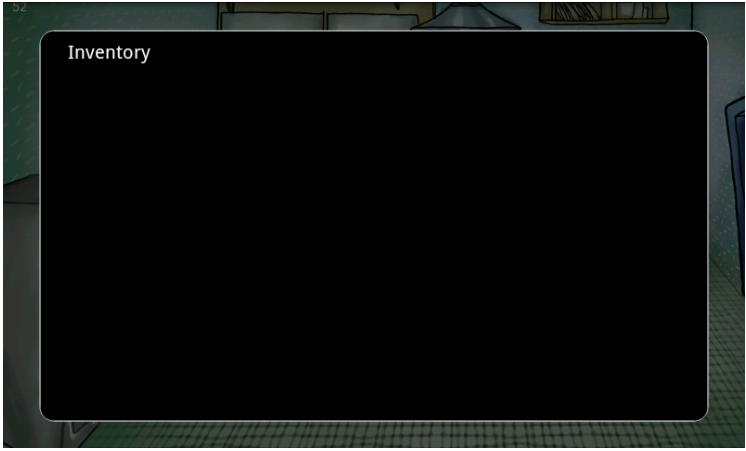
Prototipo de reproducción de video y audio

| | |
|----------------------|--|
| Titulo | Reproducción de video y audio |
| Alcance | Capacidad de reproducción de video y audio |
| Objetivo | Demostrar la viabilidad del sistema para la reproducción de video manteniendo el diseño de sistema de pintado y la reproducción simultanea de varios ficheros de audio |
| Descripción | Prototipo capaz de reproducir video en formato 3GP o MP4 que son los formatos soportados por la API Android. Además se pretende comprobar la reproducción de varios archivos de audio simultáneamente. |
| Aspectos técnicos | Debe comprobar la viabilidad de reproducción de video. Reproducción de varios archivos de audio simultáneamente de manera eficiente. Reproducción de archivos de video empaquetados en ficheros comprimidos. |
| Resultados aportados | Sí, es viable la reproducción de video manteniendo el diseño del sistema de pintado. Se da una solución a una limitación en la plataforma de Android para la reproducción de video y el pintado en un mismo objeto de tipo lienzo. También se demostró que se puede reproducir varios archivos de audio simultáneamente sin que el rendimiento se vea condicionado. La API de Android no soporta la reproducción de video desde un archivo comprimido. |

| | |
|------------------|--|
| Captura pantalla |  |
|------------------|--|

Prototipo del HUD

| | |
|-------------------|--|
| Titulo | HUD |
| Alcance | Incorporación de elementos en la interfaz grafica, sistema de eventos |
| Objetivo | Demostrar la viabilidad y funcionamiento de la creación de elementos que mejoran la interacción del usuario dentro del método de pintado manejando el sistema de eventos. |
| Descripción | Basado en los conceptos propuestos en el estudio de interacción (Véase 4.1) para dispositivos móviles se ha desarrollado un prototipo de interacción para el usuario. Este prototipo se muestra un inventario con todos los objetos que posee el personaje principal y además muestra una pequeña lupa que aparece al pulsar la pantalla táctil. |
| Aspectos técnicos | Prueba de integración del modelo de interacción propuesto en el sistema de pintado. Comprobación de la velocidad de |

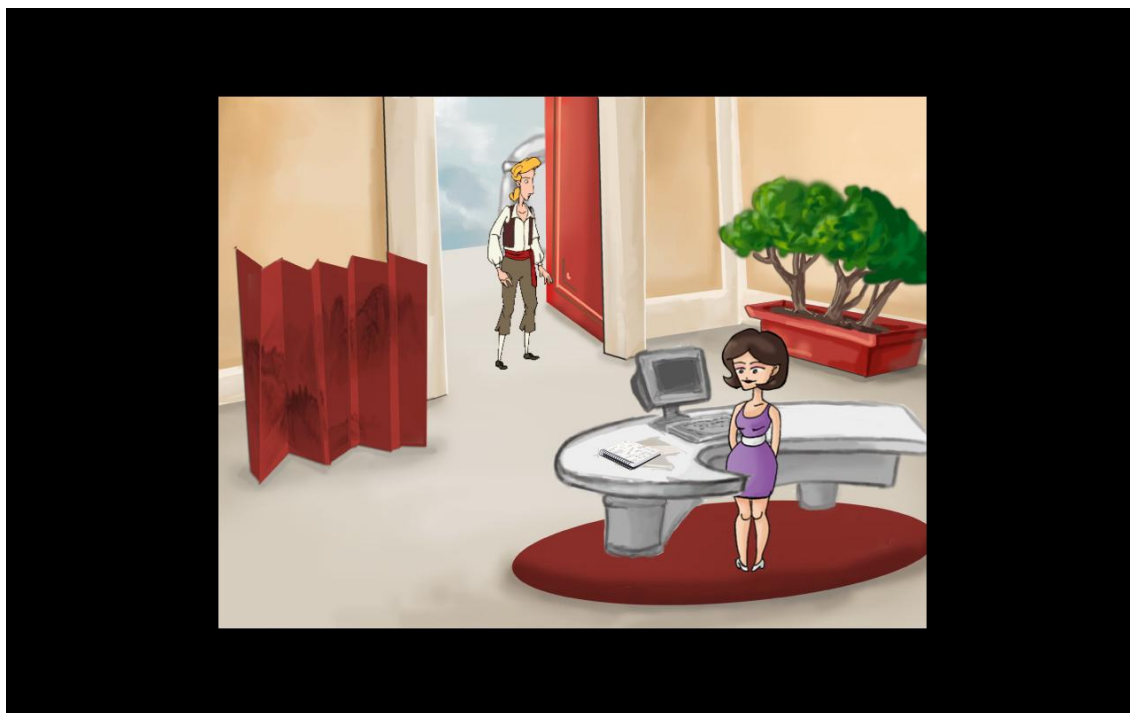
| | |
|------------|--|
| | <p>actualización al mostrarse la lupa de la interfaz. Probar las opciones del sistema de tratamiento de eventos de Android</p> |
| Resultados | <p>El funcionamiento del prototipo es bastante positivo.</p> <p>Se mantuvo el diseño del sistema de pintado original. La velocidad de refresco no se ve afectada cuando se muestra la lupa. Android facilita el reconocimiento de algunos tipos de pulsación pero la detección de las pulsaciones continuas está re-implementada. Además aunque Android suele aportar herramientas que suplen casi todas las necesidades de los desarrolladores, se ha implementado la interfaz del inventario por completo.</p> |
| Captura |  |

Compatibilidad entre las plataformas

A continuación se describen los puntos más relevantes a tener en cuenta en el análisis de la integración del soporte de Android en eAdventure, teniendo en cuenta posibles limitaciones y riesgos de compatibilidad entre ambas plataformas.

Visualización de los juegos eAdventure

La plataforma eAdventure no fue diseñada para crear juegos con distintas resoluciones, sino que se basan en una resolución estándar de 800 píxeles de ancho por 600 píxeles de alto. De esta manera al crear un juego con la plataforma eAdventure, no importa la resolución de la pantalla del ordenador, siempre se visualiza en una ventana de tamaño 800x600 píxeles.



En un ordenador de escritorio no supone una limitación importante, sin embargo si hablamos de dispositivos móviles con pantallas de tamaños reducidos, se debe aprovechar al máximo el tamaño de la pantalla sin contar con un tamaño de

visualización definido de antemano. El motor a desarrollar debe tener en cuenta este aspecto. Como solución a este cambio de visualización se propusieron dos soluciones:

- Escalar el juego al alto del dispositivo en modo apaisado. En éste modo la imagen mantiene la proporción 4:3 sin abarcar todo el tamaño del dispositivo, dejando franjas en los laterales.
- Redimensionar la interfaz al tamaño de la pantalla de forma que ocupa toda ella sin mantener la proporción original.

Por parte de la plataforma eAdventure no se tienen en cuenta más limitaciones en fase de análisis para dar soporte a múltiples pantallas.

Soporte multi-pantalla en Android

El motor de juegos a desarrollar debe visualizarse correctamente en cualquier configuración de pantalla soportada por la plataforma de Android. Para ello se trata de minimizar las modificaciones de lado del editor de la plataforma eAdventure y conseguir que los juegos creados desde éste sean compatibles con ordenadores de escritorio y a su vez con dispositivos móviles de distintas configuraciones de pantalla.

Para detectar los posibles riesgos en la visualización de los juegos se analiza el soporte de diferentes configuraciones de pantalla por parte de la plataforma Android. Además, se crean prototipos que cubren este aspecto y que aportan soluciones de diseño para dar soporte multi-pantalla al motor de juegos en Android.

Teniendo en cuenta las características de la API de Android, se decide qué tipo de configuraciones de pantalla debe soportar el motor.

Primero se deben distinguir dos partes diferenciadas del motor de juegos para Android. El cargador/gestor de juegos, con una interfaz gráfica y disposición de elementos gráficos tratados por la plataforma Android automáticamente, y el núcleo del motor, donde se ejecuta el juego y donde los elementos gráficos son ubicados y

redimensionados manualmente por el código a desarrollar. Es importante diferenciar las dos partes ya que los mecanismos de auto-adaptación de Android sólo repercuten sobre su sistema propio de disposición de elementos gráficos. De esta manera el soporte multi-pantalla en el núcleo del motor de juegos se debe hacer manualmente.

A continuación se muestran imágenes de la implementación del motor de juegos ejecutándose en distintos dispositivos móviles con pantallas de distinta resolución y densidad de píxeles

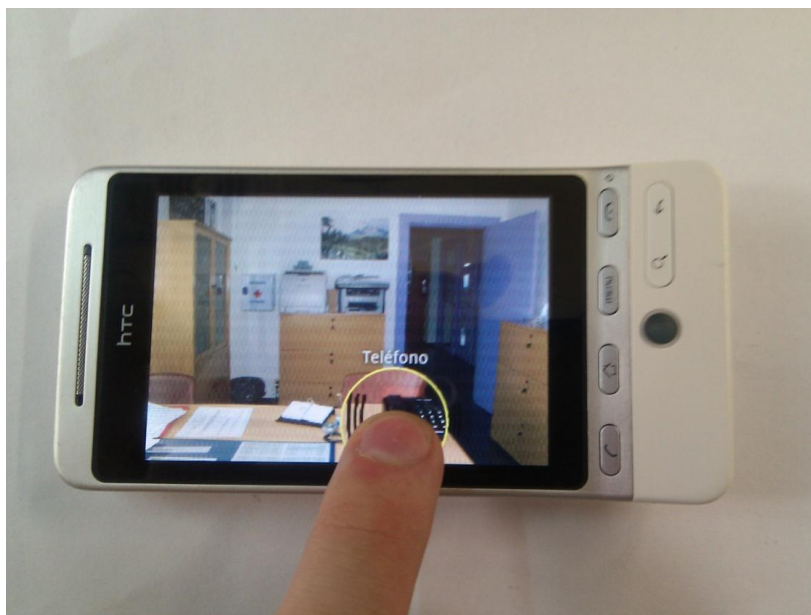


Ilustración 53 - Juego desplegado en dispositivo móvil HTC Hero. Pantalla de 3,2 pulgadas 320*480 píxeles



**Ilustración 54 - Juego desplegado en dispositivo móvil Nexus One. Pantalla de 3,7" 800*480
píxeles**



Ilustración 55 - Juego desplegado en dispositivo móvil HTC Legend. 3,2" 320*480

Formatos de fichero

Se debe tener en cuenta la compatibilidad de los formatos de fichero utilizados en la plataforma eAdventure con la plataforma Android.

A continuación se muestra una tabla comparativa de los ficheros soportados por ambas plataformas

| Formatos | eAdventure | Android |
|-----------------|-------------------|----------------|
| Bmp | SI | SI |
| PNG | SI | SI |
| Gif | SI | SI |
| Jpg | SI | SI |
| Mov | SI | NO |
| Mpg | SI | NO |
| Avi | SI | NO |
| Mp4 | NO | SI |
| Mp3 | SI | SI |
| Midi | SI | SI |
| Mid | SI | SI |

Como se puede observar hay formatos de ficheros de la plataforma eAdventure que no son soportados en la plataforma Android y viceversa. Por ello se debe hacer una elección de los recursos que van a ser compatibles entre ambas plataformas. En el caso del video no existen formatos compatibles entre ambas plataformas, con lo que se debe dar una solución a ésta incompatibilidad.

Se ha decidido dar soporte en el motor de juegos para Android a los siguientes formatos de fichero:

- Imagen: PNG ,BMP ,JPG y GIF
- Audio: MP3
- Video: MP4

En el caso del formato de video, se decide por MP4 y se debe dar una solución en la fase de autoría de eAdventure para mantener la compatibilidad con este formato de video.

7. Diseño e Implementación

El siguiente apartado tiene como objetivo mostrar los conceptos de diseño más importantes tanto del motor de juegos como de la herramienta de edición y el sistema de repositorio de juegos.

7.1. Motor de juegos para Android

A continuación se detalla el diseño en base a dos puntos de vista. En primer lugar se describe el flujo de comunicación entre los principales componentes de la interfaz de la aplicación. En segundo lugar se explica la arquitectura interna del núcleo del motor.

Componentes principales del motor

La aplicación motor de juegos está dividida en varios componentes *Activity* para cada tarea independiente de la interacción con el usuario (Véase sección 5.2). A continuación se explica la función de cada uno de éstos componentes del motor de juegos y el diagrama de flujo entre cada uno de ellos durante la ejecución de la aplicación.

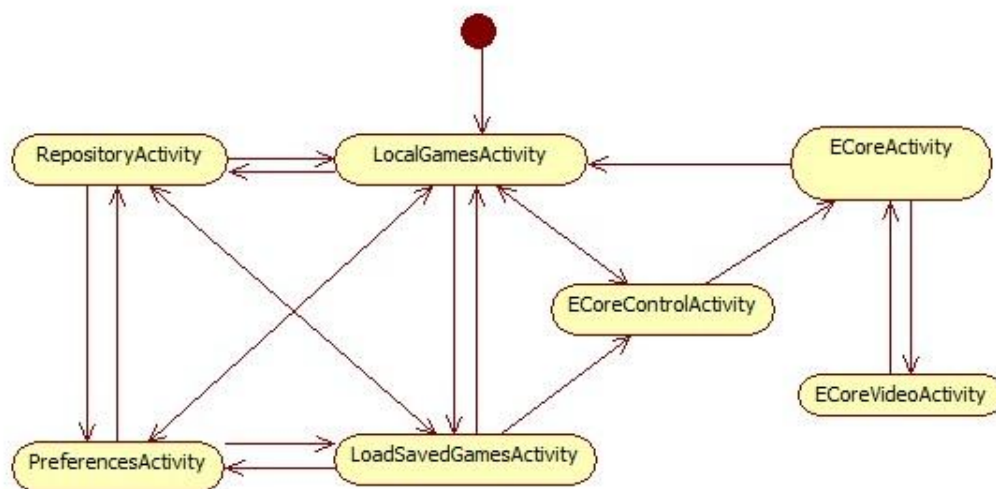


Ilustración 56 - Flujo de comunicación entre los principales componentes UML de la

LocalGamesActivity: Muestra la lista de juegos instalados y preparados para jugar. Permite comenzar la ejecución de cada uno de ellos, y además desinstalar los juegos de la lista.

RepositoryActivity: Muestra una lista con información sobre los juegos disponibles en el repositorio de juegos online. Permite descargar e instalar cada uno de ellos automáticamente.

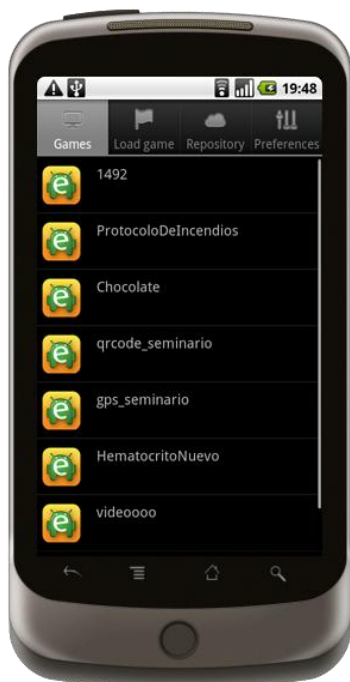


Figure 2 LocalGamesActivity



Figure 3 RepositoryActivity

LoadSavedGamesActivity: Muestra la lista de partidas guardadas previamente en un instante determinado y permite cargarlas para continuar su ejecución. Permite además eliminar las partidas guardadas.

PreferencesActivity: Muestra una lista de parámetros configurables de la aplicación como, habilitar sonidos durante el juego, activar vibración en la interacción del juego, habilitar el modo de depuración de juegos, y además permite instalar juegos ubicados en la tarjeta de memoria SD del dispositivo móvil.



Ilustración 58 - LoadGamesActivity

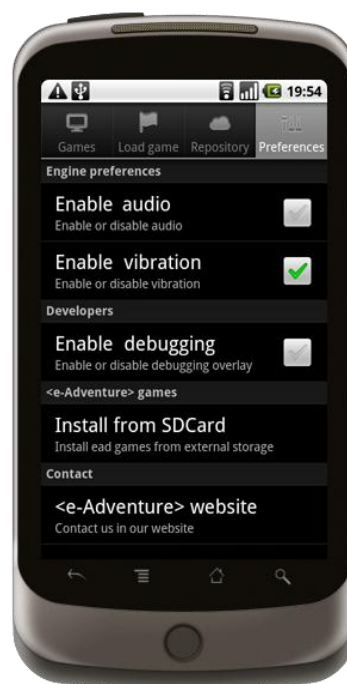


Ilustración 57 - PreferencesActivity

ECoreControlActivity: Comprueba la disponibilidad de servicios requeridos para la ejecución de ciertos juegos que utilicen geoposicionamiento (sistema receptor GPS) o reconocimiento de códigos bidimensionales (software de reconocimiento de códigos QR).

ECoreActivity: Es el componente activo durante la ejecución de los juegos. Su función es mostrar la interfaz de juego y además, recoger y transmitir las acciones del usuario al núcleo del motor (Véase Core).



Ilustración 59 - Ejecución de un juego en el componente ECoreActivity

ECoreVideoActivity: Componente encargado de la reproducción de vídeos. Debido a limitaciones impuestas por la plataforma, la reproducción de vídeo durante el juego requiere un *Activity* independiente (Véase sección Prototipo de reproducción de video y audio).

Arquitectura del motor de juegos

En ésta sección se describen los elementos más significativos de la arquitectura del motor de juegos. El siguiente esquema muestra los componentes del sistema con mayor relevancia.

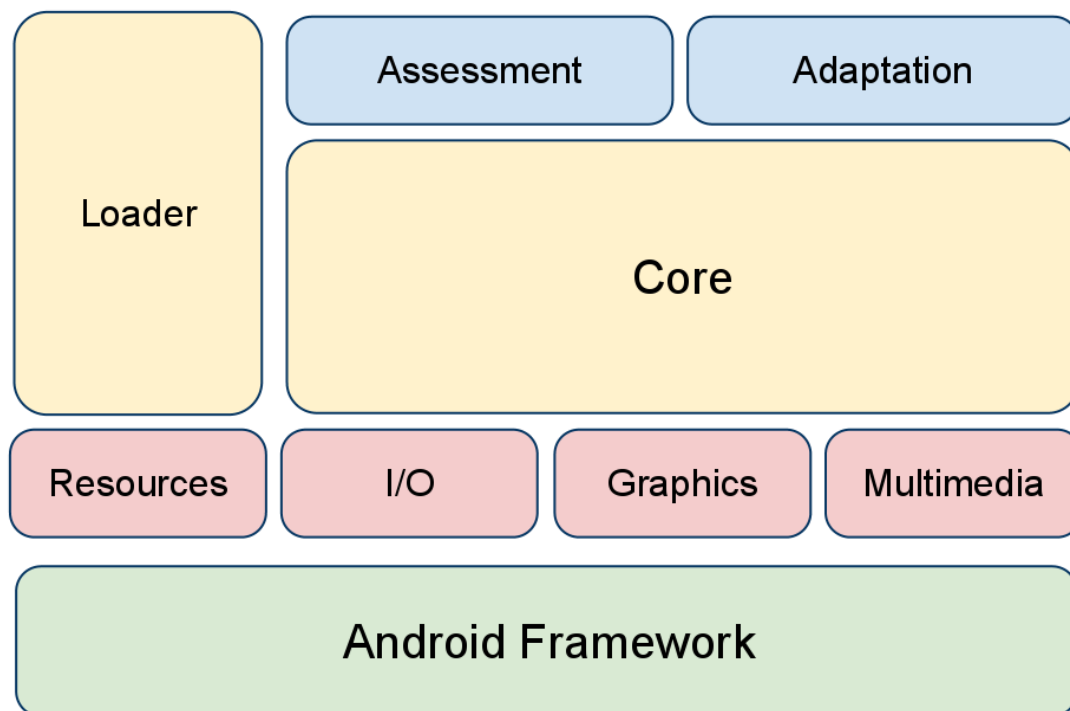


Ilustración 60 - Arquitectura interna del motor de juegos de eAdventure para Android

Loader

El componente Loader se encarga de interpretar las aventuras gráficas descritas en los ficheros de juegos. Tras realizar el análisis sintáctico de los descriptores XML del juego se crean las estructuras de datos necesarias para su ejecución. El análisis sintáctico se realiza mediante el uso de la API de SAX proporcionada por Android.

Multimedia

El componente Multimedia encargado de gestionar los recursos multimedia de los juegos. Su función consiste en cargar dichos recursos y hacerlos accesibles al resto de componentes del motor. Internamente mantiene, funcionando así como sistema de caché. De esta forma se optimiza el uso de memoria permitiendo a su vez el acceso directo a los recursos desde cualquier módulo del núcleo del motor de juegos.

Assessment

Este componente Assessment gestiona el sistema de reglas de seguimiento de la actuación del jugador durante la ejecución del juego. Para ello realiza la evaluación de dichas reglas y genera los informes correspondientes. Un informe se puede generar en los formatos XML o HTML y será mostrado tras concluir el juego. Los informes se pueden transmitir al profesor mediante correo electrónico, publicarlo en redes sociales como *Twitter* o *Facebook*, o incluso enviarlo por SMS.

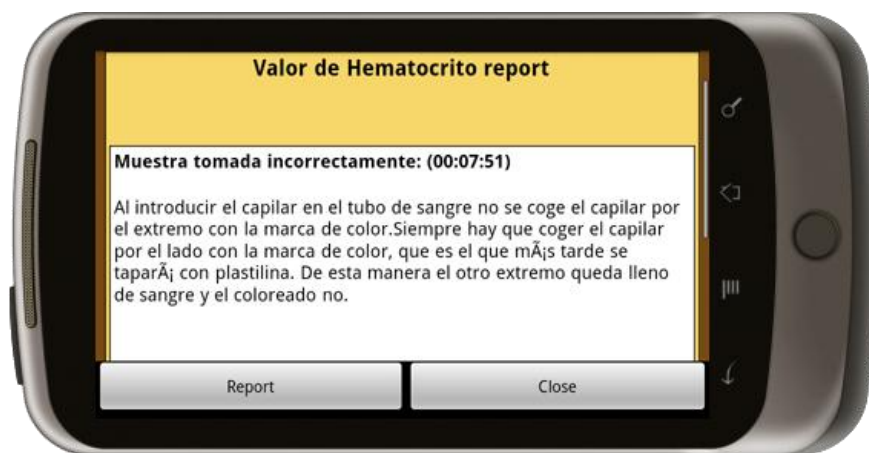


Ilustración 61 - Informe generado en componente Assessment

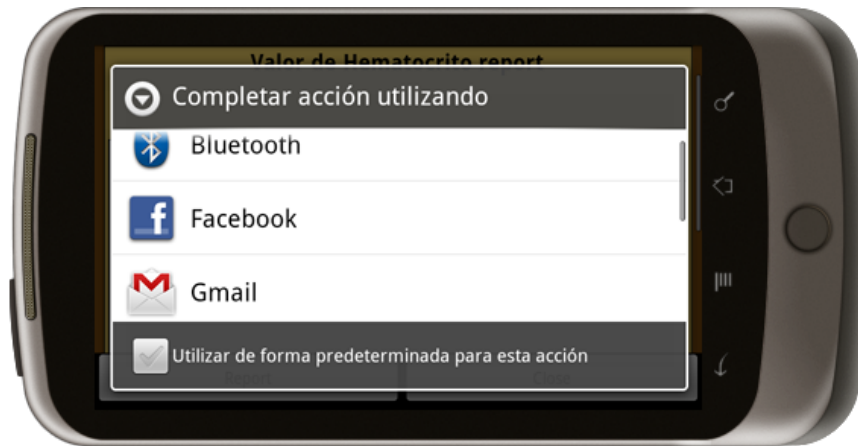


Ilustración 62 - Envío de informes de evaluación en el motor de eAdventure para Android

Adaptation

El componente *Adaptation* se encarga de gestionar la adaptación del nivel y el contenido de los juegos durante su ejecución mediante las reglas de adaptación descritas en el juego.

La inicialización y modificación de estas reglas se realiza la carga del juego.

Resources

El componente *Resources* es el encargado de acceder a la memoria secundaria del dispositivo y cargar los recursos del juegos cuando otros componentes lo requieren. Es especialmente usado por el componente multimedia.

Core

Core es el componente central del motor ya que se encarga de gestionar la lógica y los estados del juego. Su diseño está estructurado siguiendo el patrón Modelo-Vista-Controlador separado en los componentes *GUI*, *FunctionalData* y *GameControl* respectivamente:

- El componente *GUI* se encarga de representar visualmente el estado del juego. Gestiona el pintado de la escena y cada uno de los elementos presentes en ella. *GUI* además realiza el pintado de la interfaz de interacción común en todos los juegos (Véase Prototipo del método de pintado) que se encarga de mostrar el inventario del jugador o la lupa de interacción entre otros.
- El componente *FunctionalData* mantiene una estructura de datos por cada elemento funcional de un juego. Dicha estructura es creada al inicio de cada capítulo a través de la descripción obtenida por el componente *Loader*. Estos elementos contienen información como el tamaño o la referencia a los recursos gráficos de cada objeto, personaje o libro que contiene un juego.
- *GameControl* es el componente encargado de controlar el estado del juego en cada instante. Su diseño sigue los principios del patrón *State* cambiando su comportamiento en función de los distintos estados que conforman los juegos. Escenas, conversaciones, libros, escenas de diapositivas, o videos son algunos de los estados posibles durante la ejecución. Cada uno de ellos trata las acciones del usuario captadas por el sistema de eventos de Android y actualiza la estructura de datos que mantiene la lógica del juego.

7.2. Adaptación del editor de juegos de eAdventure

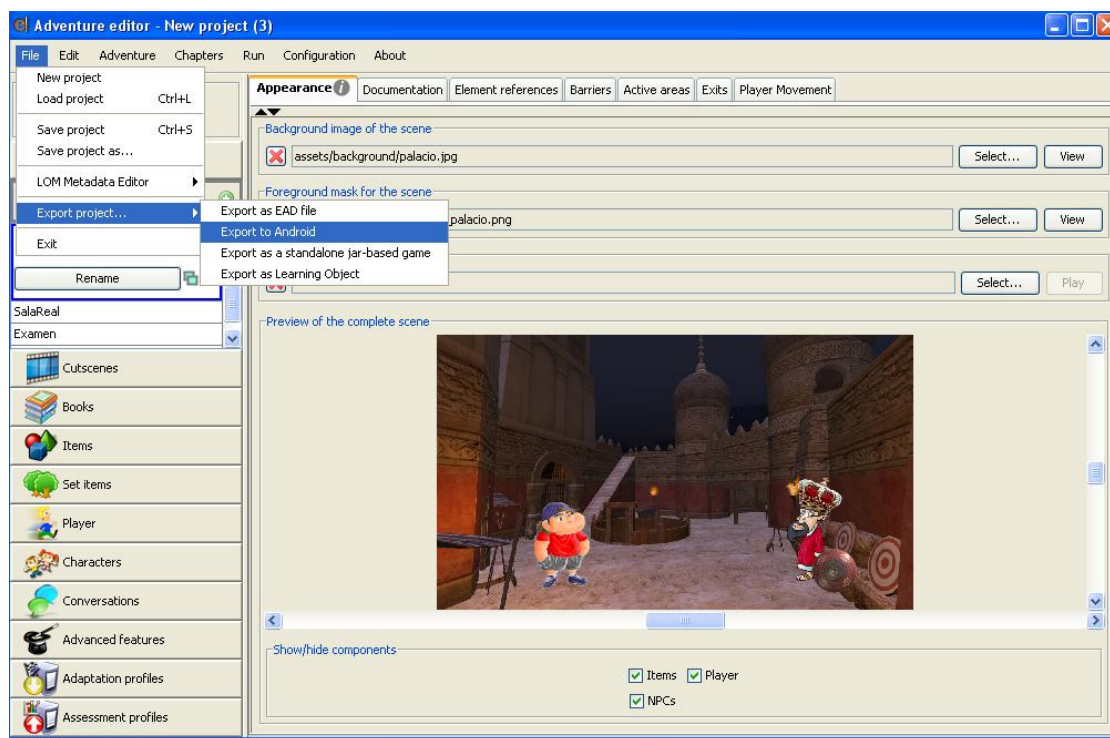


Ilustración 63 - Perfil de exportación de juegos eAdventure para Android

El editor de juegos de la plataforma eAdventure ha sido ampliado para integrar herramientas que comprueban la compatibilidad de los juegos eAdventure en dispositivos móviles. Para ello se ha implementado un sistema durante la exportación de los juegos que comprueba las siguientes medidas:

- El tamaño máximo de los recursos de cada escena debe ser inferior al límite establecido en aplicaciones Android (véase sección Detección problema de consumo de memoria).
- Los archivos de audio y video deben ser compatibles con los formatos soportados por Android (véase sección Formatos multimedia)

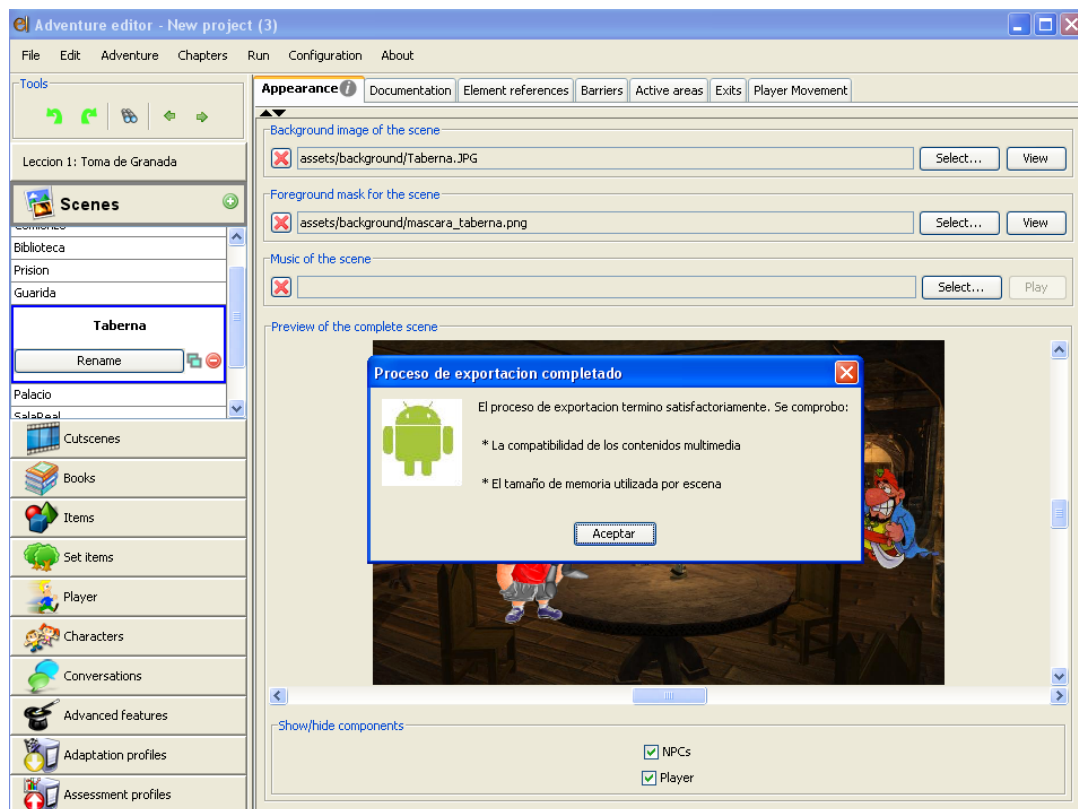


Ilustración 64 - Informe de compatibilidad en la exportación de juegos eAdventure para Android

7.3. Repositorio de juegos eAdventure

El sistema repositorio de juegos es el medio diseñado para la distribución de juegos a través de Internet. El funcionamiento de este sistema se basa en el uso de tecnologías estándar para la comunicación y la transferencia entre el servidor, que dispone de los juegos, y el cliente integrado en el motor de juegos que accede a ellos.

El repositorio que mantiene los juegos disponibles *online* está implementado a través de un servidor HTTP estándar y es accesible a través de la URL del repositorio. Dicha URL localiza un fichero XML que contiene información sobre los juegos disponibles para su descarga. La estructura de éste fichero XML se muestra a continuación.

```
<?xml version="1.0" encoding="utf-8"?>

<repository>

<game>

<title>Protocolo de Incendios</title>

<imageIcon>http://eadventure-
android.googlecode.com/files/ProtocoloDeIncendiosIcon.png</imageIcon>

<image>http://eadventure-
android.googlecode.com/files/ProtocoloDeIncendiosImage.png</image>

<description>Este juego fue creado con el objetivo de enseñar a
futuros jugadores las características del protocolo de evacuación para
casos de incendio de la Facultad de Informática de la universidad
Complutense de Madrid. Utiliza perfiles de adaptación para mejorar la
experiencia educativa.</description>

<url>http://eadventure-
android.googlecode.com/files/ProtocoloDeIncendios.ead</url>

</game>
...
</repository>
```

Cuando el usuario solicita instalar alguno de los juegos disponibles la aplicación los obtiene de la URL especificada en el XML del repositorio. El sistema de descarga está diseñado basándose en el prototipo de la sección “ Prototipo de repositorio de juegos ”, que controla la gestión de la descarga de en segundo plano.

8. Optimizaciones del sistema

Detección problema de consumo de memoria

En el proceso de depuración de la aplicación se detectó que durante la ejecución de los juegos de prueba, la aplicación se detiene y el sistema de depuración de Android reporta el siguiente error.

```
ERROR/dalvikvm-heap(3896): 6291456-byte external allocation too large for
this process.
ERROR/(3896): VM won't let us allocate 6291456 bytes
ERROR/AndroidRuntime(3896): Uncaught handler: thread main exiting due to
uncaught exception
ERROR/AndroidRuntime(3896): java.lang.OutOfMemoryError: bitmap size exceeds
VM budget at android.graphics.BitmapFactory.nativeDecodeStream(Native Method)
```

La excepción informa de un problema en la reserva de memoria de la máquina virtual durante la ejecución del proceso nativo de Android `nativeDecodeStream` dedicado a la carga de imágenes en memoria.

Por ello se decidió monitorizar el progreso del uso de memoria de la aplicación a través de la herramienta de depuración de errores incluida en el SDK de Android.

Después de varias pruebas se advirtió que el uso de la memoria en el momento de la captura de las excepciones alcanzaba un porcentaje mínimo de la cantidad total de memoria disponible “teóricamente” para las aplicaciones en Android.

NOTA: En el momento en que se realizó este testeo, las aplicaciones en Android estaban limitadas a un máximo de 16MB o 24MB de memoria en pila dependiendo de la capacidad del dispositivo en que se ejecutasen.

Tras ésta observación se investigó la forma en que la máquina virtual Dalvik gestiona la memoria y se encontró que su funcionamiento está lejos de ser intuitivo, en concreto cuando se hace uso de objetos *Bitmap*.

Gestión de objetos Bitmap en Android

La reserva de memoria de los objetos *Bitmap* es un caso especial⁶ ya que estos pueden ser cargados tanto en la pila de la máquina virtual como en la pila nativa del proceso que lo alberga, dependiendo de ciertos detalles ajenos al control del programador.

En caso de ser almacenados en la pila nativa, la reserva de memoria es gestionada directamente por el sistema operativo y no por la máquina virtual (MV). Esto implica que dichos *Bitmap*⁷ no son contemplados por el recolector de basura de la MV, y serán liberados solo cuando se eliminen todas las referencias que existan en la pila de la MV.

En cualquier caso la MV controla que la cantidad de memoria total reservada no supere el límite por aplicación. Sin embargo dicha cantidad de memoria no es reportada por la herramienta de depuración por ello es difícil de detectar.

Método para monitorizar la memoria de la aplicación

Android provee herramientas de monitorización del estado de las aplicaciones a nivel de programación. En concreto la clase *android.os.Debug* contiene métodos como *GetNativeHeapAllocatedSize()* que informa de la cantidad de memoria reservada en dicha pila nativa de la aplicación.

Haciendo uso de éstas herramientas se analizó la evolución de la memoria durante la ejecución exportando los datos recopilados en un archivo con formato CSV. Una vez obtenido el fichero se analizó usando una aplicación de hojas de cálculo y se obtuvieron gráficas como la mostrada a continuación.

⁶ <http://www.mail-archive.com/android-developers@googlegroups.com/msg16242.html> accedido el 3 de Mayo de 2010

⁷ <http://stackoverflow.com/questions/1945142/bitmaps-in-android>, accedido el 5 de Mayo de 2010

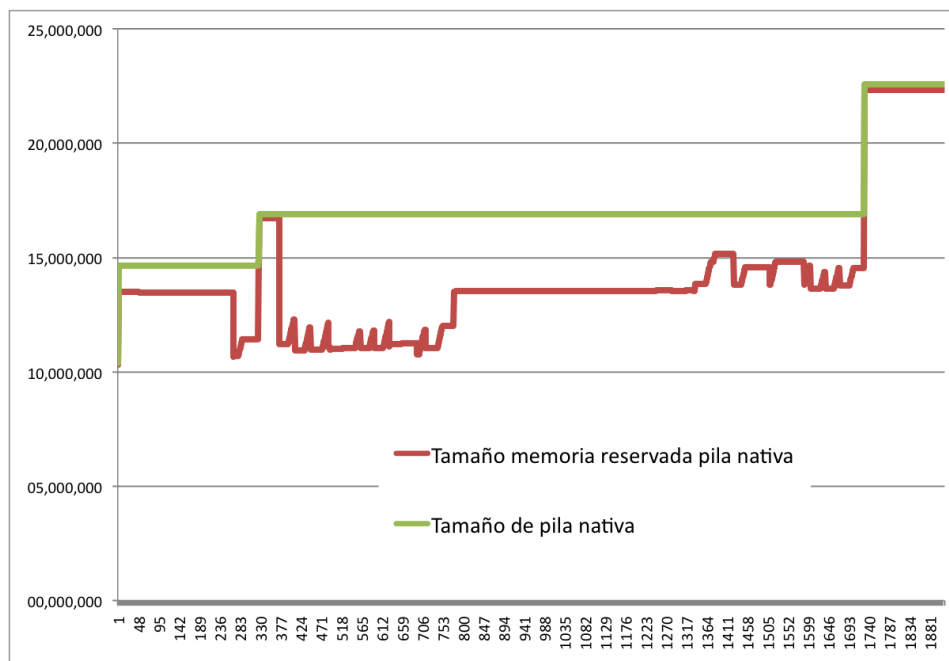


Ilustración 65 - Gráfica de consumo de memoria durante la ejecución de un juego eAdventure

Esta gráfica representa la evolución de la memoria de la pila nativa durante la ejecución de la aventura gráfica “Protocolo de incendios”. En el eje horizontal se numera cada ciclo del bucle principal del juego y en vertical se muestra la cantidad de memoria en bytes para cada variable estudiada.

Se puede observar cómo en ciertos momentos de la ejecución la reserva de memoria crece de forma notable ocasionando el aumento del tamaño total de la pila, debido principalmente a la carga de recursos gráficos. Sin embargo, se ha de mantener el tamaño de dicha pila siempre por debajo del máximo mencionado anteriormente para evitar problemas de memoria⁸.

Optimizaciones del manejo de objetos Bitmap

Tras repetir el proceso anterior con varios juegos de prueba y analizando la relación entre los datos obtenidos y el código de la aplicación, se han acotado varios puntos de el motor de juegos en los que la gestión de memoria no se estaba realizando debidamente.

⁸ <http://developer.android.com/resources/articles/avoiding-memory-leaks.html>, accedido el 6 de Mayo de 2010

Uno de los problemas encontrados es que existe en el código varias referencias a objetos *Bitmap* incluso cuando éstos no eran necesarios. Esto provoca que dichos recursos gráficos no fueran liberados.

Una solución a este problema se ha basado en cambiar el sistema caché de imágenes y eliminar redundancias en multitud de módulos que hacen uso de dichos objetos *Bitmap*. Además se sustituyeron ciertas referencias a objetos con referencias débiles (*WeakReference*) ya que estas últimas no son contabilizadas por el gestor de liberación de memoria en desuso.

```
HashMap<String, WeakReference<Bitmap>>[] imageCache;
```

Otra mejora aplicada se basa en uso de la API de Android que permite la configuración de ciertas opciones en la carga de imágenes que permite que las imágenes sean eliminadas de la memoria en caso que sea necesaria.

```
options.inPurgeable = true;  
options.inInputShareable = true
```

Depuración de juegos en el motor para Android

Con el fin de facilitar un método sencillo para monitorizar el estado de la aplicación se incluyó en el motor de juegos un modo de ejecución alternativo que muestra la gráfica del progreso del consumo de memoria superpuesta sobre los juegos en tiempo real. Además se muestran datos como el tamaño en memoria de los recursos gráficos de los elementos disponibles en cada escena, o la velocidad de refresco de la interfaz, es decir la tasa de imágenes por segundo. Esto permite analizar las causas de posibles problemas de memoria en momentos determinados de un juego. La siguiente imagen muestra un ejemplo de ejecución de juego con ésta herramienta activada.

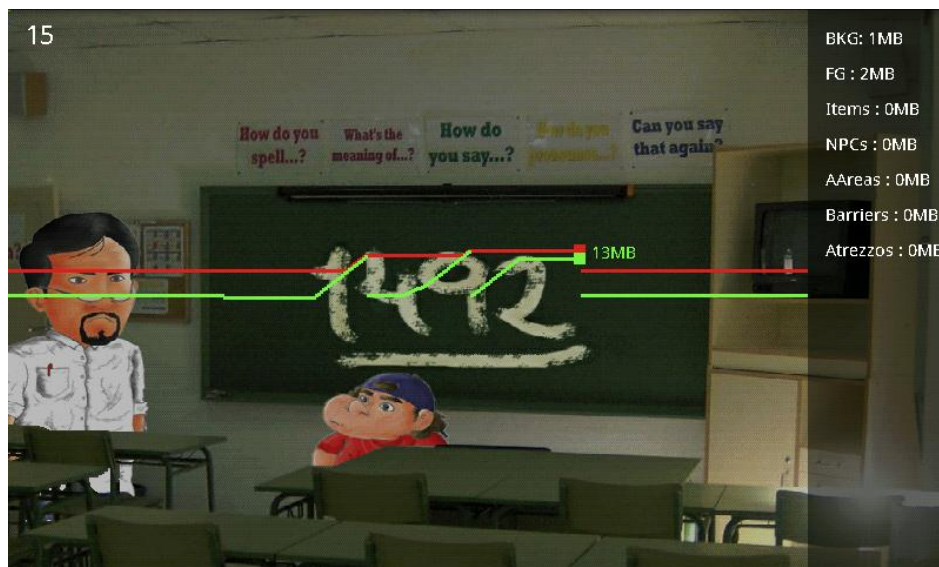


Ilustración 66 - Modo de depuración para desarrolladores en el motor de eAdventure para Android

9. Caso de uso

Durante el transcurso del proyecto se decidió realizar un prototipo de juego educativo que hiciese uso del sistema de geoposicionamiento. La idea es crear un juego educativo orientado a motivar el aprendizaje en el lugar y el momento (Just-in-time learning). El juego debe ser apto para su uso en un entorno educativo real.

Partiendo de éstas premisas se creó una aventura gráfica que sirve como complemento activo en una visita guiada durante una excursión de forma que el contenido del juego se relaciona con los lugares visitados. El juego se desarrollará en una zona determinada de Madrid y se centrará en la historia de algunos edificios emblemáticos como el Palacio Real, la catedral de la Almudena, y el Teatro Real.



Ilustración 67 - Escena inicial del juego geolocalizado “Buscando a Curro”

9.1. Descripción educativa

Valor educativo del juego

La intención del juego es despertar el interés de los alumnos por conocer la historia de la ciudad a través de la historia de sus reliquias arquitectónicas. Con el sistema de geolocalización los alumnos podrán a prueba su capacidad de orientación y localización en un mapa, así como el reconocimiento de calles y edificios para desplazarse a los lugares necesarios por el juego. Además, el juego estimula el diálogo y la colaboración entre los alumnos durante la realización de la actividad.

Escenario educativo

Esta propuesta de juego está orientada a ser un complemento educativo en excursiones o actividades extraescolares para alumnos de instituto. Obviamente debido a la temática del juego, el escenario donde se desarrollara la actividad debe estar localizado en las inmediaciones de los edificios emblemáticos destacados por el juego. Asumiendo que cada alumno dispondrá de un dispositivo móvil con Android, el tutor debe facilitar la distribución del juego a los alumnos.

Una vez que los alumnos estén situados en la localización inicial del juego, el papel del tutor es atender a los estudiantes en el transcurso del juego, como si de una excursión normal se tratase. Al finalizar, el tutor recolecta los informes con los datos de la evaluación de cada alumno, según se han definido en el juego a través de la herramienta “Assessment”.

Un modo alternativo de uso puede ser la distribución del juego, previa a la actividad de visita guiada. Esto sería una solución en caso de no disponer de dispositivos móviles compatibles. Sin embargo, pierde la ventaja de experimentar la aventura en el escenario real.

El nivel educativo al que está orientado el juego comprende un amplio rango de jugadores. Tal vez el nivel mínimo adecuado al juego corresponda con alumnos de 2º o 3º de E.S.O. basándonos en los contenidos de asignaturas como Historia.

No obstante el juego posee un rango de jugadores sin límite, y cualquier persona interesada en conocer la historia de Madrid puede encontrarlo atractivo. Un caso de uso alternativo surge como guía interactiva durante visitas turísticas a los lugares propuestos.

Evaluación del aprendizaje

Durante el transcurso del juego existen ciertas acciones que son tenidas en cuenta por el sistema de evaluación que recogerá datos como el nivel de curiosidad del jugador, tiempo dedicado, etc.

Además llegado a un punto de la historia del juego, uno de los personajes planteará una serie de preguntas referentes a los contenidos mostrados en escenas previas. El alumno deberá contestar para continuar jugando. Una vez acabado el juego, el profesor tendrá un informe individualizado con la actuación del alumno durante el juego.

9.2. Guión del juego

El juego tiene tres capítulos correspondientes a los tres edificios emblemáticos visitados de Madrid. Dependiendo de cada escena el jugador deberá interactuar con los personajes y objetos disponibles para así resolver los diferentes problemas que le serán planteados. Mientras resuelve éstos problemas, se informará de datos históricos y anécdotas que estén directamente relacionadas con la escena en la que se encuentra.

Capítulo I, El Palacio Real

“Tras escuchar gritos de auxilio el protagonista se encuentra con el rey Juan Carlos II pidiendo ayuda para encontrar al ladrón de la corona real. Pero antes de dar la pista para capturarlo, guiará al jugador por el interior del Palacio Real para mostrarle algunos datos de interés. Por último el Rey indicará la dirección por donde huyó el ladrón.”

Capítulo II, Catedral de La Almudena

“Una conversación con un sacerdote extraño sugiere que se trata del ladrón disfrazado tras haber orientado al jugador en la dirección incorrecta. Entonces otro sacerdote dará una pista interesante tras explicar la historia y arquitectura de la catedral.”

Capítulo III, el Teatro Real

“Después de que el portero del teatro real niegue la entrada, el protagonista debe conseguir un pase para acceder al interior del teatro. Para ello debe conversar con los músicos de que están junto a la fachada principal. Una vez que consigue entrar, y tras una conversación con el acomodador del teatro el juego llega a su fin con la captura del ladrón.”

9.3. Planificación y desarrollo del juego

Se decidió dividir la estructura del juego en tres capítulos independientes, para repartir el trabajo de forma equitativa. No obstante, desde el principio se estableció el argumento central de la historia y se mantuvo una línea común en todos los capítulos. Éstas son algunas de las tareas que se realizaron durante la creación del juego:

- Obtener información sobre el edificio emblemático asignado.
- Escribir el guión del capítulo correspondiente.

- Escribir los diálogos entre los personajes.
- Crear los recursos gráficos de cada escena.
- Implementar un capítulo con el editor eAdventure.

Implementación

La principal diferencia entre la creación de un juego con estas características y los juegos estándar en eAdventure es la necesidad de establecer ciertas reglas de geolocalización que dicten el funcionamiento del juego en función de la posición geográfica del jugador. No obstante, sigue siendo posible jugar a estos juegos a través de un ordenador convencional sin ningún esfuerzo adicional por parte del creador.

Para ello el sistema de condiciones y efectos de eAdventure incorpora un tipo nuevo de condiciones que permiten controlar la lógica del juego basándose en la posición geolocalizada del jugador. Se utilizan éste tipo de reglas para decidir qué escena debe mostrarse al jugador, en función de su posición.

Sin embargo esto es solo una solución aplicada a éste caso de uso, pero el sistema de condiciones permite utilizar la información de la posición del jugador para lanzar cualquier tipo de efectos (cambios de estado, acciones, etc.)

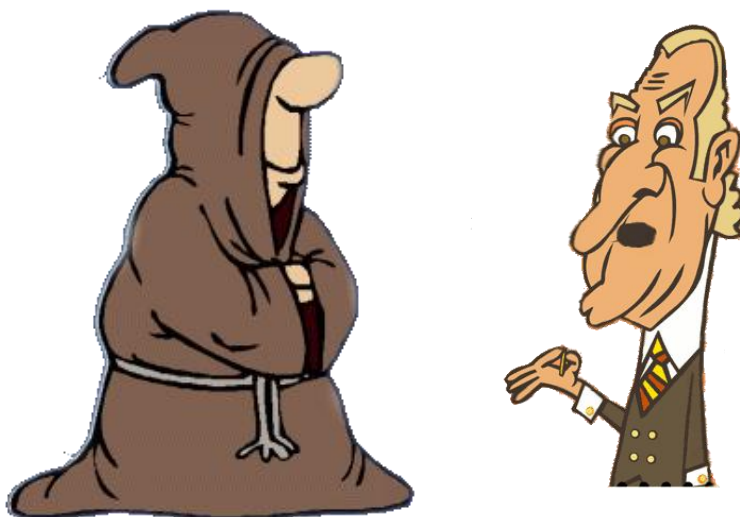
Estética y recursos artísticos del juego

Las imágenes de fondo de cada escena son fotografías reales de los lugares en los que sucede la trama. La mayoría de estas fotografías han sido obtenidas a través de la Web www.panoramio.com



Ilustración 68 - Escena durante el juego "Buscando a Curro"

Los recursos gráficos de los personajes que aparecen durante el juego fueron obtenidos independientemente de distintos servicios Web, y modificados para crear las animaciones durante el juego (hablar, andar, etc.)



Durante el juego se muestran ciertas escenas de corte con textos narrativos para orientar al jugador. Las imágenes de dichas escenas se crearon con la herramienta Microsoft PowerPoint™ y fueron exportadas como imágenes JPG independientes.

*Una mañana, en las
inmediaciones del
Palacio Real...*

Por último, en la versión para ordenador, se muestran ciertas escenas intermedias que indican la posición del protagonista y a qué lugar debe dirigirse a continuación. Para ello se hizo uso de la herramienta Google Earth™.



Documentación de referencia sobre el juego

Sobre el Palacio Real



<http://www.patrimonionacional.es/Home/Palacios-reales/Palacio-real-de-madrid.aspx>



http://es.wikipedia.org/wiki/Palacio_Real_de_Madrid

Catedral de La Almudena



<http://www.archimadrid.es/princi/princip/otros/repor/catedral/catedral.htm>



http://es.wikipedia.org/wiki/Catedral_de_la_Almudena_de_Madrid

Teatro Real



<http://www.teatro-real.es/El%20teatro/El%20edificio/Historia>



http://es.wikipedia.org/wiki/Teatro_Real_de_Madrid

10.Conclusiones

El principal objetivo del proyecto, disponer de una herramienta capaz de crear y distribuir juegos educativos en dispositivos móviles, ha sido en todo momento un objetivo ambicioso. Se debía diseñar un sistema complejo que además fuese útil en el ámbito educativo.

Por ello se decidió desarrollar el proyecto en base a una plataforma de videojuegos educativos ya existente, ampliando la funcionalidad y valor de ésta para ser aprovechada por profesores e instituciones educativas libremente. eAdventure ha proporcionado esta base sólida y experimentada con la que poder trabajar.

Por otro lado Android fue considerada la plataforma móvil más adecuada para el desarrollo del proyecto, permitiendo la libre distribución del contenido y contando con un gran número de dispositivos móviles en el mercado.

Desde el punto de vista de la investigación, el proyecto se ha centrado en dar soluciones genéricas a la adaptación de los juegos educativos en dispositivos móviles (interacción de usuario, distribución de los juegos y empleo de tecnología GPS o códigos bidimensionales).

Por otro lado, desde el punto de vista técnico se ha tenido que hacer frente a la mejora de una aplicación extensa, como es eAdventure, ampliando su funcionalidad en todo el flujo de creación, distribución y ejecución de los juegos para permitir el despliegue de los juegos en dispositivos móviles.

El desarrollo del proyecto ha tenido que afrontar distintos desafíos sin que éstos hayan afectado al alcance de los objetivos planteados inicialmente. El desarrollar sobre plataformas en constante evolución, la falta de documentación y los retos que conlleva el desarrollo de juegos para móviles son algunos de éstos desafíos

A pesar de todo se ha desarrollado una solución completa y funcional que cumple con los objetivos del proyecto. La aportación realizada facilita la creación y distribución de aventuras gráficas educativas en móviles Android, abriendo la posibilidad de crear nuevas formas de jugar motivadas por el uso de la posición global del jugador y la

interacción con el entorno real a través del uso de la cámara de fotos y los códigos bidimensionales.

10.1. Contribuciones

Tras el trabajo realizado podemos afirmar que el principal objetivo del proyecto ha sido completado con éxito proporcionando una plataforma libre, totalmente funcional y capaz de crear y distribuir juegos educativos en dispositivos móviles. Prueba de ello son las contribuciones que han sido realizadas directamente sobre la plataforma de eAdventure y en el ámbito educativo.

Adaptación juegos educativos a dispositivos móviles

A lo largo del desarrollo del proyecto se han estudiado y documentado distintas fórmulas de adaptación de juegos educativos, creados para ordenadores de escritorio, a dispositivos móviles. A continuación se muestran las principales contribuciones asociadas a ésta línea de actuación.

Interacción de usuario: Se ha estudiado la interacción en juegos de tipo aventura gráfica desplegados en móviles teniendo en cuenta el paso de un modelo de interacción basado en *point&click*, usado en ordenadores de escritorio (teclado y ratón), a un modelo de presionar y arrastrar con el dedo sobre una pantalla de móvil táctil y de tamaño reducido.

Distribución de los juegos: Se han estudiado distintas fórmulas de distribución de juegos educativos haciendo uso de las capacidades de conectividad de los terminales móviles de última generación.

- Distribución haciendo uso de Internet (Descarga directa, a través de la tienda de aplicaciones de la plataforma móvil, basada en repositorio)

- Distribución directa a través de la conexión del dispositivo a ordenadores de escritorio (USB)
- Distribución de juegos entre terminales a través de conectividad de campo cercano (*Wifi* o *Bluetooth*)

GPS y Cámara de Fotos (Códigos QR): Se han estudiado distintas formas de juego posibles en videojuegos educativos de tipo aventuras gráfica haciendo uso de los sensores de geoposicionamiento (GPS) y el reconocimiento con la cámara de fotos de códigos bidimensionales QR

Herramienta de autoría

En la parte de edición de juegos se ha mantenido el mismo proceso con el que contaba la plataforma eAdventure. Se ha añadido soporte para perfiles de exportación a Android en la que se realiza una comprobación de compatibilidad de los recursos utilizados en el juego para correr en Android. De esta manera el desarrollador puede desplegar los juegos que genere en plataforma de escritorio, Web o móvil Android.

Por último, se ha mantenido la compatibilidad con los juegos creados en versiones previas por lo que se pueden editar y exportar en móviles sin necesidad de modificaciones.

Distribución

Para facilitar el despliegue de los juegos se han implementado distintas opciones de distribución de estos para su uso en distintas situaciones.

- Sistema de repositorio de juegos: implementado haciendo uso de tecnologías estándar (HTTP ,XML) y accesible desde la aplicación de Android

- Instalación de juegos ubicados en la memoria del móvil: Posibilita la instalación de juegos eAdventure en el móvil a través de la memoria externa de manera que se puede transferir el fichero de juegos a través del cable del móvil, *Bluetooth* o descarga directa de Internet y posteriormente instalarlo.

Ejecución de juegos educativos

Se ha creado un motor de juegos eAdventure para dispositivos móviles Android dando soporte a todas las características de los juegos creados con la plataforma.

Este permite la ejecución de aventuras de primer y tercera persona basadas en escenas, escenas de corte, video elementos y objetos , objetos de atrezzo personajes y protagonista , libros.

- Permite la ejecución de juegos creados previamente para el motor de escritorio.
- Se ha dado soporte a juegos basados en la localización del jugador (geolocalizados) y juegos que hacen uso de la cámara de fotos como medio de interacción con el entorno real a través de códigos QR.
- Se ha integrado el sistema de informes de evaluación de eAdventure con la posibilidad de publicarlos como es habitual a través de e-mail o a través de distintas redes sociales como Facebook, Twitter o Buzz.
- Se ha tenido en cuenta la compatibilidad con distintos dispositivos, permitiendo la ejecución de juegos en dispositivos con pantallas de distintos tamaños.
- Se han implementado distintas formas de instalación de juegos, a través del repositorio de descargas o instalación directa desde memoria interna del dispositivo (*Bluetooth* , descarga directa y transferencia directa).
- La aplicación motor permite el salvado y cargado de partidas, de manera que el jugador puede parar la partida en cualquier momento.
- Se ha incorporado una herramienta para desarrolladores que permite observar en tiempo real el consumo de memoria y la velocidad de refresco de los juegos con el fin de detectar posibles inefficiencias en el uso de recursos.

10.2. Trabajo futuro

Durante el transcurso del proyecto han surgido nuevas líneas de interés para tener en cuenta en futuros trabajos. Por otro lado la aplicación desarrollada deja algunos aspectos por mejorar. A continuación describimos posibles ampliaciones que consideramos interesantes para ser incluidas en versiones futuras de la plataforma.

Autoría

Geoposicionamiento y códigos QR en el editor

La ejecución de juegos con Geoposicionamiento y juegos con códigos QR es viable en el motor de juegos para dispositivos móviles. Sin embargo, como trabajo futuro resulta interesante pensar una manera intuitiva y sencilla de poder adaptar el editor para poder crear este tipo de juegos, como por ejemplo, vinculando las escenas de los juegos con puntos sobre un mapa o la generación de códigos QR automáticamente por el editor, para liberar de trabajo al profesor.

Compatibilidad recursos multimedia

Por el momento el editor cuenta con soporte para distintos tipos de archivos multimedia. Sin embargo, Android no soporta todos esos formatos debido a las limitaciones de la plataforma, con lo que sería interesante dar una solución en el editor para poder transformar los ficheros multimedia no soportados por Android a ficheros soportados sin que el profesor usuario percibiese ningún cambio y no

Distribución

Integración con gestores e-Learning

Aun habiendo implementado un sistema de distribución de juegos versátil, como es el repositorio de juegos, se debe considerar la importancia de los sistemas gestores de contenido orientados a la educación (LMS). Por lo tanto sería interesante mejorar el sistema, integrándolo con alguno de éstos gestores como por ejemplo *Moodle* o *WebCT*.

Ejecución

Nuevas fórmulas de interacción

Aun realizando un estudio de la interacción de usuario y dando una solución general para los distintos dispositivos móviles Android, se considera importante facilitar otros tipos de interacción más específicos de cada dispositivo móvil haciendo uso de distintas tecnologías como el *trackball*, teclado físico o el sensor de acelerómetro.

Accesibilidad

En el motor actual de eAdventure para Android se incluye la posibilidad de sintetizar la voz de las conversaciones. En todo caso, se considera interesante ampliar la accesibilidad del motor en un futuro con el fin mejorar la jugabilidad de los juegos eAdventure para las personas con discapacidad visual. La plataforma de Android dispone de herramientas en el framework de desarrollo que dan soporte a la Accesibilidad permitiendo la implementación de soluciones totalmente integradas con la plataforma.

Reflexión

Habiendo finalizado el proyecto nos sentimos satisfechos por el trabajo realizado y por alcanzar los objetivos planteados desde su comienzo.

Los retos que se plantearon inicialmente, tales como desarrollar para dispositivos móviles haciendo uso de nuevas tecnologías o hacer contribuciones a una plataforma libre de juegos educativos extensa y en continua evolución, no ha supuesto una barrera para llevar a cabo el proyecto.

Además ha sido una experiencia muy valiosa para el desarrollo profesional de los integrantes del proyecto ya que se han adquirido conocimientos técnicos pertenecientes a distintas ramas de la informática y se ha aprendido a planificar , gestionar y desarrollar un proyecto software por completo.

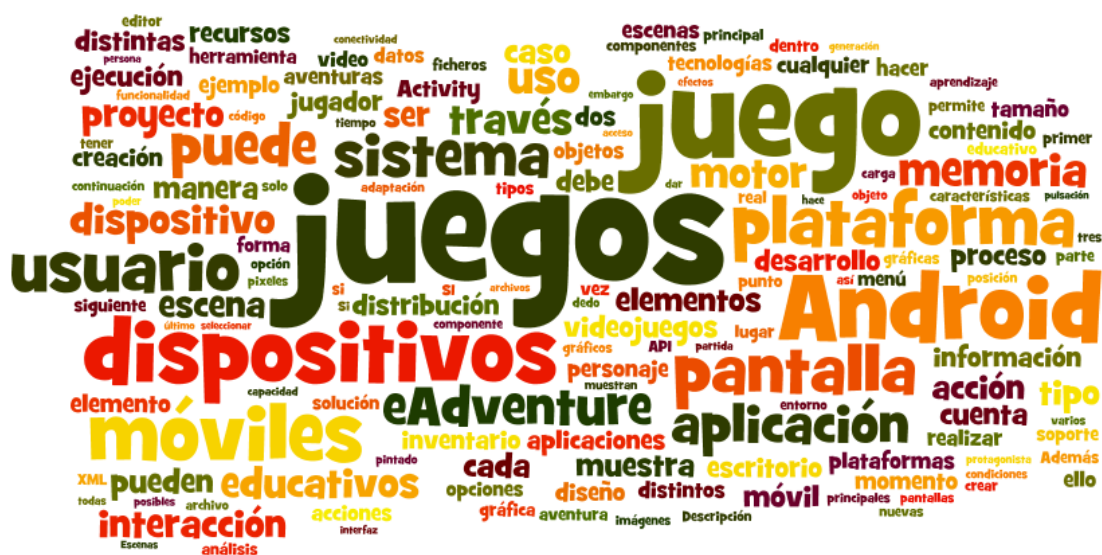


Ilustración 69 - Nube de palabras de la memoria del proyecto

11.Referencias

M Dunlop, S Brewster, *The challenge of mobile devices for human computer interaction*, Personal and Ubiquitous Computing, 2002

R. V. Eck, Digital Game-Based Learning: It's not just the digital natives who are restless, EDUCASE Review, vol. 41, no. 2, 2006

J Gong, P Tarasewich, *Guidelines for handheld mobile device interface design*, Proceedings of DSI 2004 Annual Meeting, 2004

Hay, K.E. 1999. *The digital weather station: A study of learning with with 5D visualization*. Paper presented at the Annual meeting of the American Educational Research Association, Montreal, Canada

12. Bibliografía

- Grupo de desarrolladores de Android.
<http://groups.google.com/group/android-developers>
- Documentación oficial de la plataforma Android
<http://developer.android.com/index.html>
- Foro de desarrollo de aplicaciones para Android
<http://stackoverflow.com/questions/tagged/android>
- J. Torrente, P. Moreno-Ger, I. Martínez-Ortiz, B. Fernández-Manjón, "Integration and Deployment of Educational Games in e-Learning Environments: The Learning Object Model Meets Educational Gaming," *Educational Technology & Society*, vol. 12, n. 5, pp. 359-371, 2009.
- M.D. Kickmeier-Rust, N. Peirce, O. Conlan, D. Schwarz, D. Verpoorten, and D. Albert, "Immersive Digital Games: The Interfaces for Next-Generation ELearning,"
- *Universal Access in Human-Computer Interaction. Applications and Services*, vol. LNCS 4556, 2007.
- J. Kemp, D. Livingstone, and P. Bloomfield, "SLOODLE: Connecting VLE tools with Emergent Teaching Practice in Second Life," *British Journal of Educational Technology*, vol. 40, pp. 551-555, 2009.
- Henrik Kniberg, Scrum and XP from the Trenches, Jun 27, 2007
- "Essential Facts About the Computer and Video Game Industry," Entertainment Software Association (ESA) 2009.

- J. Torrente, Á. del Blanco, E.J. Marchiori, P. Moreno-Ger, and B. Fernández-Manjón, "<e-Adventure>:Introducing Educational Games in the Learning Process," in *IEEE EDUCON 2010* Madrid, Spain: IEEE, 2010.
- P. Moreno-Ger, I. Martínez-Ortiz, J.L. Sierra, and B.Fernández-Manjón, "A Content-Centric Development Process Model.," *IEEE Computer*, vol. 41, pp. 24-30, 2008.
- A. Litchfield, L. Dyson, E. Lawrence, and A Zmijewska, "Directions for m-learning research to enhance active learning," in *ASCILITE - ICT: Providing choices for learners and learning*, Singapore, 2007, pp. 587-596.
- T. Andrews, R. Caladine, and R. Smyth, "Utilizing Students' Own Mobile Devices and Rich Media: Two Case Studies from the Health Sciences," in *Second International Conference on Mobile, Hybrid, and Online Learning*, Saint Maarten, Netherlands, Antilles, 2010, pp. 71 - 76.
- S. Hsi, "A study of user experiences mediated by nomadic web content in a museum," *Journal of Computer Assisted Learning*, vol. 19, pp. 308-319, 2003.
- M. Ruchter, B. Klar, and W. Geiger, "Comparing the effects of mobile computers and traditional approaches in environmental education," *Computers & Education*, vol. 54, pp. 1054–1067, 2010.
- M. Kam, V. Rudraraju, A. Tewari, and J. Canny, "Mobile Gaming with Children in Rural India: Contextual Factors in the Use of Game Design Patterns," in *3rd Digital Games Research Association International Conference*, Tokyo, Japan, 2007.

- S. Natkin, C. Yan, S. Jumpertz, and B. Market, "Creating Multiplayer Ubiquitous Fames Using an Adaptive Narration Model Based on a User's Model," in *Digital Games Research Association International Conference (DiGRA 2007)*, Tokio, Japan, 2007.
- K.L. Schrier, "Revolutionizing history education: Using augmented reality games to teach histories," *Cambridge, MA.: Massachusetts Institute of Technology*, 2005.
- J. Doswell and K. Harmeyer, "Extending the 'Serious Game' Boundary: Virtual Instructors in Mobile Mixed Reality Learning Games," in *Digital Games Research Association International Conference (DiGRA 2007)*, Tokio, Japan, 2007.
- J. Lave and E. Wenger, *Situated learning: Legitimate peripheral participation*, 1991.
- E. Klopfer, K. Squire, and H. Jenkins, "Augmented reality simulations on handheld computers," in *2003 American Educational Research Association Conference Chicago, IL, USA*, 2003.
- E. Dieterle, C. Dede, and K. Schrier, "'Neomillennial' learning styles propagated by wireless handheld devices," in *Ubiquitous and pervasive knowledge and learning management: Semantics, social networking and new media to their full potential*, M. Lytras and A. Naeve, Eds. Hershey, PA: Idea Group, Inc., In press.
- Kinshuk, J. Suhonen, E. Sutinen, and T. Goh, "Mobile Technologies in Support of Distance Learning," *Asian Journal of Distance Education*, vol. 1, pp. 60-68, 2003.

- L. Lunce, "Simulations: Bringing the benefits of situated learning to the traditional classroom," *Journal of Applied Educational Technology*, vol. 3, pp. 37-45, 2006.
- P. Lavín-Mera, J. Torrente, P. Moreno-Ger, and B. Fernández-Manjón, "Mobile Game Development for Multiple Devices in Education," in *4th International Conference on Interactive Mobile and Computer-Aided Learning Amman*, Jordan, 2009.